

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
BACHARELADO EM ENGENHARIA ELÉTRICA**

EMERSON ANDREY DA CONCEIÇÃO

PEDAL DE EFEITOS DIGITAL DESTINADO A GUITARRA ELÉTRICA

JOINVILLE

2022

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
BACHARELADO EM ENGENHARIA ELÉTRICA**

EMERSON ANDREY DA CONCEIÇÃO

PEDAL DE EFEITOS DIGITAL DESTINADO A GUITARRA ELÉTRICA

Trabalho de Conclusão de Curso submetido ao Bacharelado em Engenharia Elétrica do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, para a obtenção do Grau de Engenheiro Eletricista.

Orientador: Prof. Dr. Aleksander Sade Paterno

JOINVILLE

2022

EMERSON ANDREY DA CONCEIÇÃO

PEDAL DE EFEITOS DIGITAL DESTINADO A GUITARRA ELÉTRICA

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Elétrica como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Banca Examinadora

Orientador:

Prof. Dr. Aleksander Sade Paterno
UDESC

Membros:

Prof. Dr. Eduardo Bonci Cavalca
UDESC

Prof. Dr. Fabrício Noveletto
UDESC

Joinville, 4 de Fevereiro 2022.

A minha família

AGRADECIMENTOS

Agradeço primeiramente a Deus.

Agradeço também a toda minha família, especialmente a minha mãe Andreia, por todo o apoio, cuidado e carinho durante essa jornada.

Agradeço a minha namorada Paula e toda a sua família pelo apoio incondicional.

Agradeço a todos os meus amigos que ajudaram e contribuíram com todo o meu conhecimento adquirido seja na engenharia ou na vida

*Lave o rosto nas águas sagradas da pia, nada como
um dia após o outro dia (Racionais MC's)*

RESUMO

Este trabalho apresenta o projeto de um pedal de efeitos digital destinado à guitarra elétrica. Utilizando um processamento em tempo real e o microcontrolador STM32F407VGT6, este trabalho demonstra como processar sinais de áudio de forma eficiente. Além disso, um estudo sobre captadores magnéticos é realizado, procurando caracterizar os mesmos. A partir desse estudo, é projetado circuitos analógicos com o objetivo de condicionar o sinal de áudio. Por fim, algoritmos em linguagem C são desenvolvidos com o intuito de criar quatro efeitos de guitarra: Distorção, *Delay*, *Flanger* e *Tremolo*.

Palavras-chave: Processamento digital de sinais. STM32F407VGT6, Efeitos de guitarra. Distorção. *Delay*. *Flanger*. *Tremolo*.

ABSTRACT

This work presents the project of a digital guitar effects pedal destined to electric guitar. Using real-time processing and the microcontroller STM32F407VGT6, this work shows how to process audio in a efficient way. Beyond that, a study about magnetic pickups is done, aiming characterize the same. From that study, it's designed analog circuitry conditioning the audio signal. Lastly, algorithms in C language are designed to create four electric guitar effects: Distortion, Delay, Flanger and Tremolo.

Keywords: Digital Signal Processing. STM32F407VGT6. Guitar Effects. Distortion. Delay. Flanger. Tremolo

LISTA DE FIGURAS

2.1	Captador magnético	16
2.2	Tipos de guitarra	17
2.3	Circuito equivalente	18
2.4	Resposta em frequência para valores de L e C	20
2.5	Banda passante dos captadores MG-32	21
2.6	Valor da indutância do captador na frequência	22
2.7	Valor da tensão de saída	22
3.1	Sistema de aquisição de dados	24
3.2	Circuito captador - pré amplificador	25
3.3	Pré-amplificador para condicionamento do sinal	26
3.4	Conversor AD PCM1802	29
3.5	Espectros dos sinais	31
3.6	Resposta em frequência do filtro <i>Butterworth</i>	32
3.7	Projeto do filtro <i>Butterworth</i> 5ª Ordem	34
3.8	Análise do primeiro estágio	35
3.9	Análise do primeiro estágio	36
3.10	Análise filtro de segunda ordem <i>Sallen -Key</i>	37
3.11	Sinal no domínio do tempo para o filtro anti <i>aliasing</i>	39
3.12	Tensão após o filtro <i>antialiasing</i> com senoide de entrada de 24kHz	39
3.13	Resposta em frequência apenas do fitro	40
3.14	Circuito para o filtro de reconstrução	41
3.15	Pinos conversor DA	42
3.16	Placa escolhida	43
3.17	Circuitos de alimentação	45
3.18	Sistema projetado	46
3.19	Placa de circuito impresso analógica	47
4.1	Arquitetura interna do pedal	48
4.2	Fluxograma do <i>firmware</i>	49
4.3	Protocolo de comunicação i^2S	50
4.4	Arquitetura DMA - STM32F407	52
4.5	<i>Double buffer</i>	53
4.6	<i>Double buffer</i> na prática	55
5.1	Distorção por Araya e Suyama (1996)	58

5.2	Distorção <i>hard clipping</i>	58
5.3	Distorção com corte em 50% da entrada	59
6.1	Diagrama de blocos para <i>delay</i> FIR	62
6.2	Resposta em frequência para o efeito <i>Delay</i>	63
6.3	Resposta ao impulso de um filtro pente	64
6.4	Buffer circular	67
7.1	Efeito <i>flanger</i> por gravador de fitas	68
7.2	Representação em diagramas de bloco <i>flanger</i>	69
7.3	Resposta em frequência filtro <i>comb</i> com atraso variável	71
7.4	<i>Buffer</i> com dois ponteiros	72
7.5	Interpolação linear	73
8.1	Diagrama de blocos para efeito tremolo	76
8.2	Transformada de <i>Fourier</i> dos sinais	77
8.3	Forma de onda para o sinal modulado e a portadora	78
8.4	Forma de onda para a FFT do sinal modulado com <i>zoom</i>	78
8.5	Forma de onda para a FFT do sinal modulado sem <i>zoom</i>	78
9.1	Sinais de entrada e saída	80
9.2	<i>FFT</i> do sinal de saída	81
9.3	Atuação dos filtros analógicos	81
9.4	Corrente consumida	82
9.5	Algoritmo de Distorção	82
9.6	<i>FFT</i> do sinal de entrada	83
9.7	<i>FFT</i> do algoritmo de distorção	83
9.8	Resposta ao degrau - <i>Delay</i>	85
9.9	<i>Double buffer</i> - <i>Delay</i>	86
9.10	Resposta ao degrau - <i>Flanger</i> com máxima variação	87
9.11	Resposta ao degrau - <i>Flanger</i> com mínima variação	88
9.12	<i>Double buffer</i> para a rotina <i>Flanger</i>	88
9.13	Efeito tremolo no domínio do tempo	89
9.14	Efeito Tremolo no domínio da frequência	90
9.15	<i>Double buffer</i> para o efeito <i>Tremolo</i>	90
A.1	Pinagem do microcontrolador	97
A.2	Regulador, Proteção e chave de liga/desliga	98
A.3	Pré-amplificador e filtro passa-baixa	99
A.4	Filtro de reconstrução	100
A.5	Placa de circuito impresso	101

LISTA DE TABELAS

2.1	Valores dos parâmetros do circuito de um captador	19
3.1	Parâmetros PCM1802	29
3.2	Conexões do conversor AD	30
3.3	Resumo dos parâmetros filtro <i>anti-aliasing</i>	32
3.4	Valores dos componentes do filtro	34
3.5	Valores dos componentes do filtro	41
3.6	Conexões ADPCM1802	42
3.7	Periféricos do microcontrolador	43
3.8	Banda mínima para os AmpOps	45
4.1	Parâmetros DMA	52
6.1	Configuração do DMA	65

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Estrutura	13
2	REVISÃO BIBLIOGRÁFICA - GUITARRA	15
2.1	Captadores	15
2.2	Princípio de funcionamento	17
2.3	Circuito equivalente de um captador	18
2.3.1	Frequências da guitarra	20
2.3.2	Guitarra abordada nesse projeto	20
2.4	Síntese do capítulo	23
3	HARDWARE PARA O PROJETO	24
3.1	Condicionamento do Sinal	24
3.2	Projeto pré-amplificador	25
3.2.1	Circuito de polarização	26
3.2.2	Impedância de entrada	27
3.2.3	Ganho do estágio de condicionamento	27
3.2.4	Análise em frequência do pré-amplificador	28
3.3	Conversor AD	28
3.4	Filtro anti <i>aliasing</i> - Teoria	30
3.4.1	Estágio um	34
3.4.2	Estágio dois	35
3.4.3	Estágio três	38
3.5	Filtro <i>antialiasing</i> - Simulação	38
3.6	Filtro de reconstrução	40
3.7	Conversor DA	41
3.8	Microcontrolador	43
3.9	Regulador de tensão	44
3.10	Amplificadores Operacionais	45
3.11	Síntese do capítulo	46
4	SOFTWARE PARA O PROJETO	48
4.1	<i>Código Principal</i>	48

4.2	Protocolo de comunicação	50
4.3	Controlador DMA	51
4.4	<i>Double buffer</i>	53
4.5	Síntese do capítulo	55
5	DISTORÇÃO	56
5.1	Processamento não linear	56
5.2	Modelos	57
5.2.1	<i>Waveshaping</i>	57
5.2.2	Modelo digital	59
5.3	Síntese do capítulo	60
6	DELAY	61
6.1	Análise Qualitativa	61
6.2	Análise Quantitativa	62
6.2.1	Armazenamento	64
6.2.2	<i>Buffer</i> circular	65
6.3	Síntese do capítulo	67
7	FLANGER	68
7.1	Oscilador - LFO	73
7.2	Síntese do capítulo	75
8	TREMOLO	76
8.1	Simulação	77
8.2	Síntese do capítulo	79
9	RESULTADOS	80
9.1	Distorção	82
9.2	Delay	84
9.3	Flanger	86
9.4	Tremolo	89
9.5	Síntese do capítulo	91
10	CONCLUSÃO	92
	REFERÊNCIAS BIBLIOGRÁFICAS	93
	ANEXO A – Esquemático PCI - Microcontrolador e pinagem	97

1 INTRODUÇÃO

Para um guitarrista ter impacto em uma banda, é necessário uma unidade de efeito capaz de modificar o som da guitarra, reconhecido popularmente como “pedal de guitarra”. Os efeitos de guitarra começaram a ser populares na década de 1960 (WELCH; WRIGHT; MORROW, 2006). De fato, uma das primeiras unidades que proporcionavam os efeitos nas guitarras foi o *Fuzz Tone Pedal*, datado de 1962, e considerado um dos primeiros pedais de guitarra distribuídos amplamente (WEBSTER; LENOST; KLANG, 2013).

Segundo Welch, Wright e Morrow (2006), como estes pedais eram inicialmente analógicos, então cada pedal proporcionava apenas um tipo de efeito. Dessa forma, um guitarrista que precisasse de mais de um efeito, necessitaria de vários tipos de pedais. Porém, cada pedal tinha sua particularidade: nível de tensão de alimentação; cabos de conexão e chaves para ligar e desligar o equipamento, o que tornava o uso de vários efeitos um pouco caótico.

No entanto, com o surgimento do processamento digital de sinais, rapidamente, os pedais analógicos foram substituídos pelos digitais, dessa forma, um único músico poderia ter vários recursos em apenas um pedal (WELCH; WRIGHT; MORROW, 2006).

Neste projeto, pretende-se analisar e projetar um pedal de guitarra digital em tempo real. Inicialmente, será feito uma análise sobre os captadores da guitarra, visando entender como o som característico da guitarra é criado e transmitido, em seguida, serão projetados circuitos de condicionamento de sinais, para que o microcontrolador utilizado possa criar os efeitos desejados. Com o sinal adequado, pretende-se analisar e projetar algoritmos em linguagem C, para criar os efeitos. O pedal deverá possuir os mesmos requisitos que os pedais comerciais, como tensão de operação (9V), impedância de entrada alta e impedância de saída baixa. O projeto final consiste em uma placa, com todos os circuitos necessários descritos anteriormente.

1.1 ESTRUTURA

Os capítulos deste projeto estão descritos a seguir.

Capítulo dois: Apresenta o princípio de funcionamento de um captador de guitarra, evidenciado a aplicação direta de leis da física. Além disso, o capítulo apresenta dados básicos para o projeto, como banda passante do sistema, tensão e impedância de entrada.

Capítulo três: Apresenta o desenvolvimento de filtros e condicionamento do sinal visando a parte eletrônica.

Capítulo quatro: Apresenta a arquitetura interna do sistema projetado, assim como periféricos usados dentro do microcontrolador.

Os capítulos **cinco, seis, sete e oito** apresentam, separadamente, o estudo e projeto dos efeitos digitais.

2 REVISÃO BIBLIOGRÁFICA - GUITARRA

A guitarra é um instrumento derivado do violão. Ambos são instrumentos de cordas que utilizam a vibração proveniente da mesma para gerar o som. As cordas do violão, por si só, não são capazes de produzir ondas sonoras que podem ser distinguidas claramente. Dessa forma, o violão em conjunto com a vibração da corda possui um corpo oco que se unem para vibrar e amplificar a onda sonora que surge através da corda (JUNGMANN, 1994).

O surgimento da guitarra elétrica provém da necessidade de transmitir as ondas sonoras com maior amplificação em diferentes ambientes. Alguns autores datam o surgimento das mesmas na década de 30, (JUNGMANN, 1994), ou na década de 50 (MOHAMAD; DIXON; HARTE, 2017), em virtude do surgimento do *Jazz* e outros gêneros musicais que faziam apresentações para grandes públicos (HUNTER, 2018).

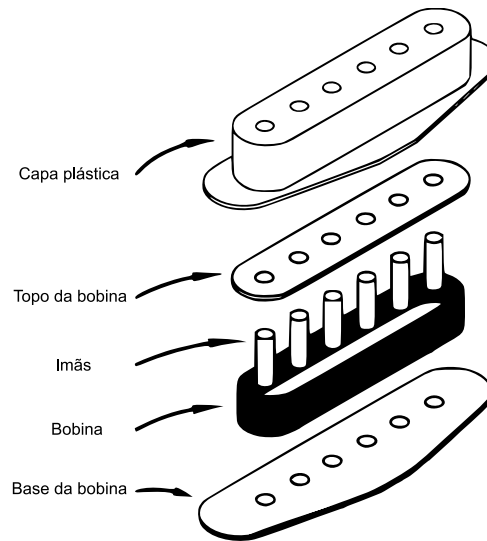
Com essa necessidade, criou-se então os captadores. Os captadores são chamados de transdutores, pois convertem a frequência de vibração das cordas em energia elétrica (TIAN CHANG SU, 2018). Esses componentes são os mais importantes no instrumento, pois o som da guitarra depende, majoritariamente, da montagem dos mesmos. Alguns parâmetros como espessura da bobina, número de voltas, tipo de ímã afetam diretamente o timbre do instrumento (TIAN CHANG SU, 2018). Outros fatores importantes são a posição e o tipo de captadores que se têm em uma guitarra.

2.1 CAPTADORES

Os captadores, inicialmente, eram constituídos de basicamente um ímã permanente e uma bobina enrolada ao seu redor (MILLS, 2016). Com o passar dos anos, ao invés de utilizar apenas uma barra de ímã, optou-se por utilizar seis peças de ímã diferentes para captar diretamente o som de cada uma das seis cordas, de forma distinta (JUNGMANN, 1994). Os captadores descritos anteriormente são chamados de captadores magnéticos, e somente conseguem captar ondas provenientes de cordas metálicas, devido a lei de Faraday. Existem outros tipos de captadores, chamados de piezoelétricos, que são capazes de captar ondas sonoras de qualquer tipo de corda, como por exemplo, a corda de *nylon*. Dessa forma, os piezoelétricos tem sua utilização majoritariamente em violões, enquanto que em guitarras têm-se os captadores magnéticos (TIAN CHANG SU, 2018).

A figura 2.1 mostra a montagem de um captador magnético.

Figura 2.1 – Captador magnético



Fonte: Adaptado de (JUNGMANN, 1994)

A bobina na figura 2.1 é constituída geralmente de fio AWG 42, 43 e 44 (HUNTER, 2018). O imã mais comum é do tipo AlNico 5, que é uma combinação entre alumínio, níquel e cobalto. Existem também imãs do tipo cerâmico. As diferenças entre esses imãs podem ser reconhecidas no timbre dos sons captados e na amplitude da tensão de saída dos mesmos. Usualmente, os captadores cerâmicos possuem sons mais agudos (com maior frequência) e mais sensíveis, gerando uma tensão de saída maior (JUNGMANN, 1994). A base dos captadores varia entre fabricantes, sendo o material mais comum o plástico.

Existem dois tipos de captadores: *Single-Coils* e *Humbuckers*. Os captadores do tipo *Single-Coil* recebem esse nome pois possuem apenas uma bobina ao redor das peças de imãs, enquanto que os do tipo *Humbucker* possuem duas bobinas.

A figura 2.2 mostra duas entre as guitarras mais famosas que utilizam esses dois tipos de captadores. A guitarra da esquerda (vermelho e laranja) é a chamada Les Paul e possui captadores do tipo *Humbucker*. Usualmente, esses captadores possuem maior amplitude da tensão de saída, em virtude de possuir duas bobinas para captar o som, e sua criação partiu do problema do qual os captadores agiam como antena, sofrendo interferência eletromagnética. Com o advento desses captadores, esse ruído é eliminado simplesmente conectando as bobinas em série ou paralelo, porém com inversão de fase (JUNGMANN, 1994).

A guitarra da direita (azul claro) é uma guitarra do tipo *Stratocaster* da *Fender*. Esse tipo de guitarra possui três captadores do tipo *single-coil* (uma bobina ao redor dos imãs). Os três captadores recebem seu nome em função da posição. Sendo eles *Neck*, *Middle* e *Bridge* em relação a figura 2.2, de cima para baixo.

Figura 2.2 – Tipos de guitarra



Fonte: (GIBSON, 2020) e (FENDER, 2020).

2.2 PRINCÍPIO DE FUNCIONAMENTO

O funcionamento de um captador é baseado na lei de Faraday. Segundo Ramos (2016), o campo elétrico em um sistema eletromagnético (com fontes variáveis no tempo) é composto por duas parcelas: conservativa e dependente do potencial elétrico e outra não conservativa, o qual se relaciona com a força eletromotriz induzida pela a variação do fluxo magnético.

Expressando a lei de Faraday com apenas a parcela não conservativa, pode-se obter a tensão induzida nos terminais da bobina do captador, representada na equação (2.1) (JUNGSMANN, 1994) e (TIAN CHANG SU, 2018).

$$U_m = -N \frac{d}{dt} \int_s B \cdot dS \quad (2.1)$$

Na equação acima, N representa o número de voltas da bobina, e a integral da indução magnética B pode ser expressa como na equação (2.2)

$$U_m = -AN \frac{dB}{dt} = -N \frac{d\phi}{dt} \quad (2.2)$$

Onde $\phi = B.A$. A equação (2.2) é a equação final da tensão sobre a bobina. Considerando que a bobina possui uma resistência R, pode-se achar a corrente apenas dividindo a expressão anterior por R (TIAN CHANG SU, 2018).

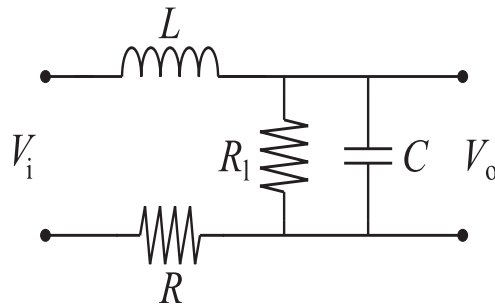
O imã permanente cria ao seu redor um fluxo magnético. Com as cordas inicialmente estacionárias, o fluxo magnético é constante, dessa forma, sua derivada é nula e não existe tensão induzida nas bobinas dos captadores. Uma vez que as cordas metálicas são tocadas, elas interagem com o fluxo magnético do imã alterando as linhas de fluxo. Dessa forma, a derivada não é nula e existe uma tensão induzida. A frequência com que a corda vibra é a mesma frequência que é captada e transmitida através de cabos e amplificadores.

Captadores que estão localizados próximos da ponte (*bridge*), ou seja, perto dos controles de volume e tom, possuem uma resposta em frequência mais "brilhante", o que significa que possuem uma banda de frequência maior, enquanto que captadores perto do braço da guitarra possuem uma resposta em frequência mais "escura", representando uma banda de passagem com frequências mais baixas (PAIVA; PAKARINEN; VÄLIMÄKI, 2012) e (JUNGMANN, 1994).

2.3 CIRCUITO EQUIVALENTE DE UM CAPTADOR

O circuito equivalente de um captador pode ser visto na figura 2.3.

Figura 2.3 – Circuito equivalente



Fonte: (PAIVA; PAKARINEN; VÄLIMÄKI, 2012)

Os parâmetros L , C , R e R_l da figura 2.3 caracterizam a banda de passagem de um captador. A função de transferência, segundo Paiva, Pakarinen e Välimäki (2012) é dada pela equação a seguir.

$$\frac{V_o(s)}{V_i(s)} = H_c(s) = \frac{1}{s^2 LC + s\left(\frac{L}{R_l} + RC\right) + \left(1 + \frac{R}{R_l}\right)} \quad (2.3)$$

Onde a indutância L é derivada do imã, a resistência R é derivada da resistência da bobina e também é conhecida como resistência DC, pois basta medir com um ohmímetro a resistência nos terminais do captador para encontrá-la. A capacitância C é impedância formada entre cada fio, e R_l é as perdas magnéticas do imã.

Alguns valores dos parâmetros podem ser obtidos na tabela a seguir.

Tabela 2.1 – Valores dos parâmetros do circuito de um captador

L (H)	R (kΩ)	R_l (kΩ)	C (pF)
1 à 10	5 à 15	300 à 3000	15 à 200

Fonte: (PAIVA; PAKARINEN; VÄLIMÄKI, 2012).

Substituindo $s = j\omega$ na equação (2.3), tem-se a transformada de *Fourier* da função de transferência.

$$H_c(j\omega) = \frac{1}{-\omega^2 LC + j\omega\left(\frac{L}{R_l} + RC\right) + \left(1 + \frac{R}{R_l}\right)} \quad (2.4)$$

Fazendo com que ω aproxime-se de zero, a equação (2.4) pode ser expressa como a seguir.

$$\lim_{\omega \rightarrow 0} H_c(j\omega) = \frac{1}{\left(1 + \frac{R}{R_l}\right)} = \frac{R_l}{R + R_l} \quad (2.5)$$

A tabela 2.1 mostra que R_l é muito maior que R. Dessa forma, quando ω tende a zero, a função de transferência tende a um.

Fazendo com que ω aproxime-se do infinito, a equação (2.4) pode ser expressa como a seguir.

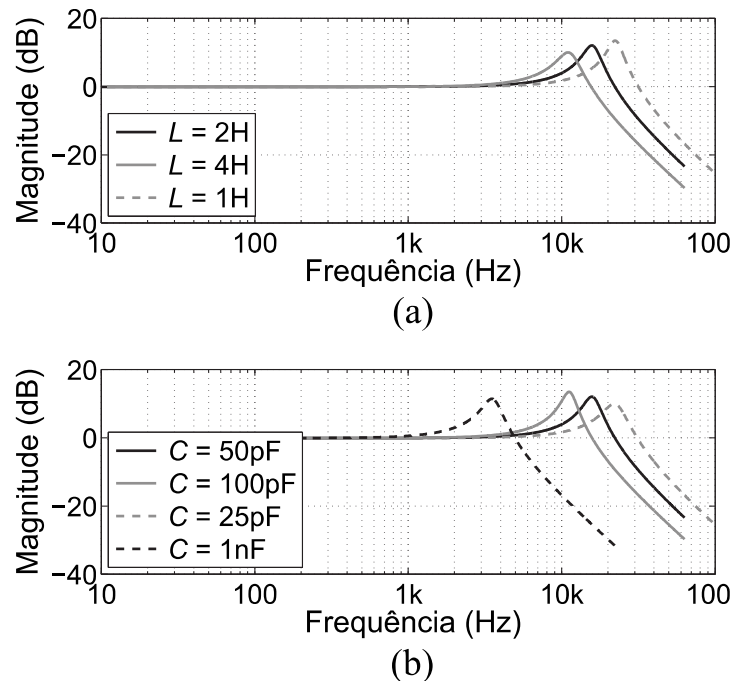
$$\lim_{\omega \rightarrow \infty} H_c(j\omega) = \frac{1}{\omega^2 LC} = 0 \quad (2.6)$$

As últimas duas equações mostram que os captadores possuem uma características de filtros passa baixas, com frequência de ressonância $\omega_R = \frac{1}{\sqrt{LC}}$ (PAIVA; PAKARINEN; VÄLIMÄKI, 2012). Tal fato pode ser observado na figura 2.4.

Dessa forma, cada fabricante pode produzir diferentes timbres para captador, variando apenas seus parâmetros elétricos. A classificação dos captadores depende do gosto musical do ouvinte, geralmente, são produzidos transdutores que possuam uma resposta constante na faixa audível, e cada ouvinte modifica essa resposta com pedais de efeitos ou circuitos de *tone*.

Os circuitos de controle de tom (*tone*) são filtros passivos compostos por resistores e capacitores que alteram a banda do sinal. São do tipo passa-alta e passa-baixa e variam de acordo com o modelo da guitarra.

Figura 2.4 – Resposta em frequência para valores de L e C



Fonte: Adaptado de (PAIVA; PAKARINEN; VÄLIMÄKI, 2012)

2.3.1 Frequências da guitarra

Segundo Bertoldi (2016), a banda passante de uma guitarra varia entre 65Hz e 6kHz, considerando um aumento de duas oitavas através de um pedal de efeito.

Uma guitarra sem pedal de efeitos possui banda passante de 82,4Hz à 1,047kHz, considerando a afinação padrão, isto é, a 6ª corda é afinada em mi, assim como a primeira (COSTA, 2020).

Dessa forma, fica evidente que a construção de um sistema embarcado deve assegurar-se de ter uma faixa de frequência entre 82,4Hz e 1,047kHz, para que, seja possível reproduzir o sinal da guitarra sem atenuação e preservar o timbre característico do instrumento.

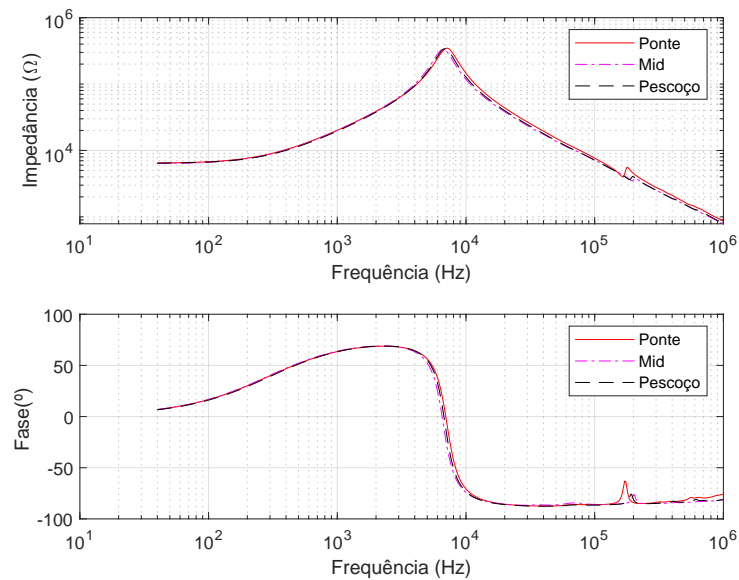
2.3.2 Guitarra abordada nesse projeto

Para este projeto, a guitarra utilizada é a MG-32 da marca *Tagima*. Essa é uma guitarra de preço baixo e comum entre músicos iniciantes. O modelo dessa guitarra segue o padrão *stratocaster*, com três captadores *single-coils*, dois controles de *tone* e um de volume.

A figura a seguir mostra o módulo da impedância e a fase dos três captadores em relação a frequência. Com isso, pode-se saber a frequência de ressonância do captador, caracterizando a sua banda de passagem.

A análise do captador foi realizada no analisador de impedâncias *agilent 4294A*. O procedimento foi feito entre as frequências de 40Hz e 1MHz. Com os dados obtidos, pode-se plotar a impedância e a fase, ambas em relação a frequência. Os três captadores possuem características praticamente idênticas, isso se deve ao fato do produto ser de baixo custo e, conseqüentemente, de baixa qualidade. Geralmente, os captadores para posição (*mid*, *neck* e *bridge*) possuem características diferentes.

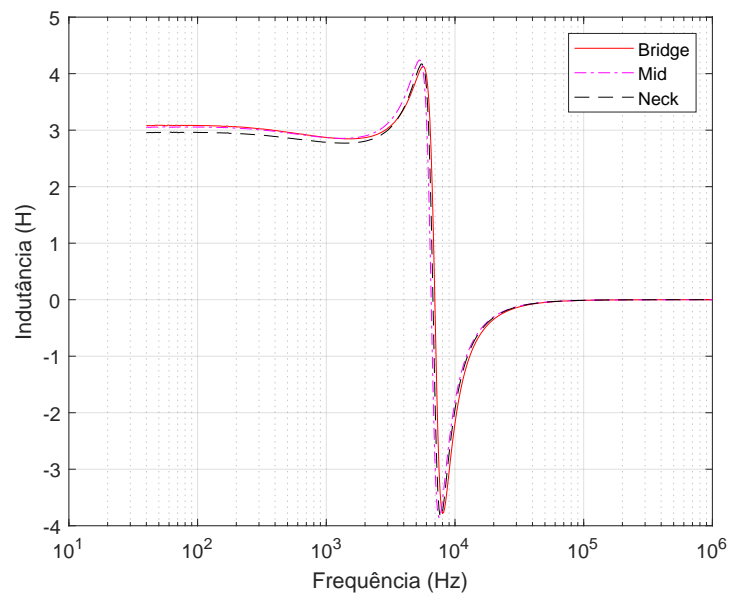
Figura 2.5 – Banda passante dos captadores MG-32



Fonte: Produção do próprio autor

Os valores das indutâncias para os captadores da guitarra M3-32 deste estudo são praticamente as mesmas, aproximadamente, $3H$ para todas as frequências abaixo da ressonância, que será a banda de frequência utilizada neste trabalho. A figura a seguir mostra o valor da indutância na frequência.

Figura 2.6 – Valor da indutância do captador na frequência

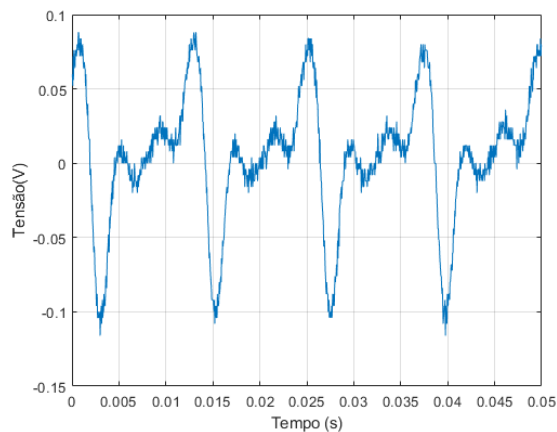


Fonte: Produção do próprio autor

A tensão de saída do captador depende de vários fatores, alguns deles comentados na seção 2.2. Porém, existem outros, como por exemplo a força aplicada na corda, quanto maior a força chamada de "ataque", tem-se uma tensão maior. Para *single-coils*, a tensão de saída é geralmente 100mV de pico (DAILEY, 2011).

A tensão de saída pode ser vista a seguir. Para a guitarra MG-32, a tensão de saída foi medida utilizando um osciloscópio. Para 6ª corda afinada em mi (82,4Hz). Nota-se que a tensão de saída não excede 100mV.

Figura 2.7 – Valor da tensão de saída



Fonte: Produção do próprio autor

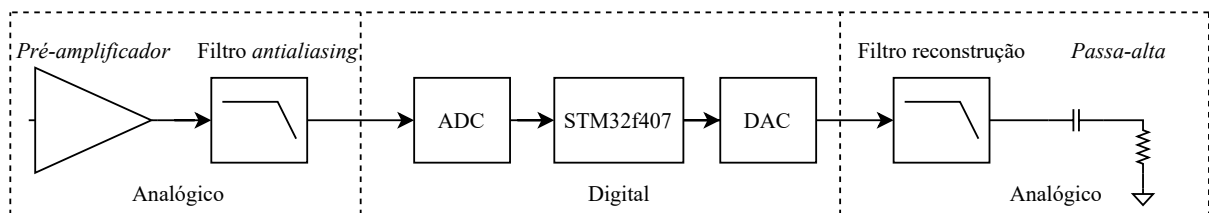
2.4 SÍNTESE DO CAPÍTULO

Deve-se entender o comportamento da guitarra ao longo do espectro audível para que modificações no seu sinal não alterem a sua característica. Esse capítulo serve como base para o capítulo seguinte, relacionando a faixa de frequências da guitarra MG-32 com a frequência de corte dos filtros necessários (comentados a seguir). Além disso, demonstra que o captador é uma fonte de tensão variável com baixa capacidade de entregar potência. Dessa forma, demonstra que um circuito de condicionamento deve ser projetado para que ao conectar o pedal junto ao captador, o pedal não sirva como carga para o mesmo.

3 HARDWARE PARA O PROJETO

Neste capítulo será abordado o *Hardware* necessário para o projeto que consistirá em uma placa de circuito impresso com todos os circuitos projetados. A figura a seguir demonstra um diagrama de blocos para um sistema de aquisição de dados.

Figura 3.1 – Sistema de aquisição de dados



Fonte: Produção do autor

O coração do projeto está situado no meio digital (amostragem e quantização do sinal sonoro). Entretanto, deve-se haver um bom condicionamento do sinal, para que os ruídos não sejam digitalizados. De forma geral, a faixa de frequência da guitarra não deve ser alterada pelos circuitos analógicos listados neste capítulo.

Os capítulos que se seguem vão descrever os blocos da figura anterior. Além disso, o circuito de alimentação será descrito também nesse capítulo (o circuito foi omitido da figura).

3.1 CONDICIONAMENTO DO SINAL

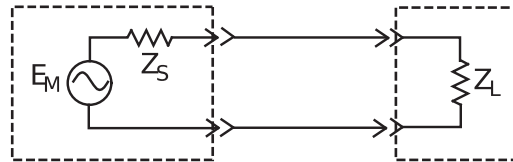
A primeira parte do projeto consiste na elaboração de um pré-amplificador. Tal circuito aparece em qualquer aparelho de áudio comercial e serve para amplificar o sinal da guitarra, propiciar um ganho constante durante a banda de passagem do áudio e uma alta impedância de entrada.

No capítulo anterior, as frequências de ressonâncias dos captadores e a tensão de saída foram medidas. Essas informações são importantes para o projeto do pedal. Como comentado, a tensão de saída é em média 100mV de pico. Essa tensão deve ser amplificada para que o conversor AD possa digitalizar o sinal, sem que haja ruídos.

Portanto, é necessário amplificar o sinal. Também, deve-se aplicar um nível DC na tensão do captador, uma vez que, para este sistema embarcado, não haverá alimentação negativa para os AmpOs, utilizados visando uma praticidade no projeto e ancorando-se em padrões comerciais que utilizam apenas um bateria ou fonte de alimentação de 9V.

Para que a informação de um captador/microphone seja preservada, é necessário que a impedância do circuito conectado ao mesmo seja muito maior. O circuito entre um captador e um pré amplificador é mostrado a seguir.

Figura 3.2 – Circuito captador - pré amplificador



Fonte: adaptado de (BALLOU, 2008)

É evidente que o circuito é um divisor de tensão. A impedância Z_S refere-se ao captador, e a impedância Z_L refere-se ao pré-amplificador. O valor da impedância do pré-amplificador influencia diretamente no fidelidade do sinal.

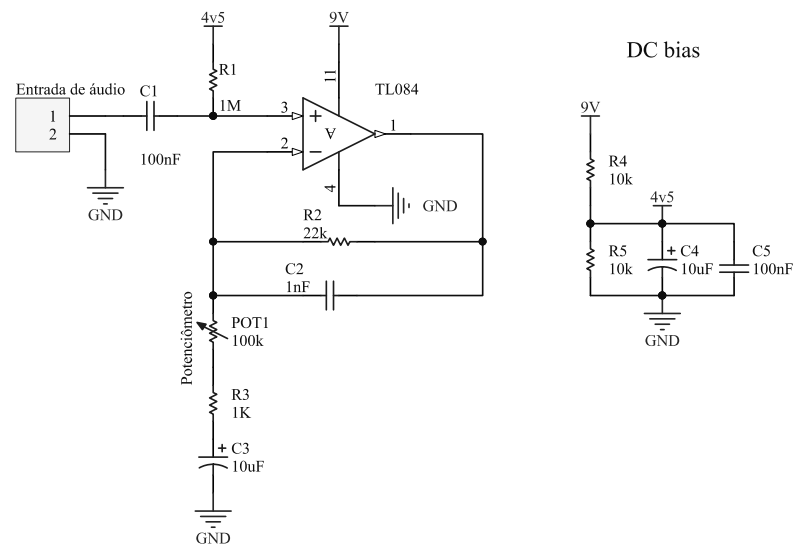
O objetivo deste circuito é utilizar um valor de Z_L muito maior do que Z_S para que, praticamente, todo o sinal de tensão possa ser enviado ao sistema (BALLOU, 2008). Valores típicos de Z_L estão na faixa de $M\Omega$. Alguns pedais de guitarra famosos mundialmente utilizam $1M\Omega$ como impedância de entrada (BOSS, 2020).

Para amplificar um sinal de tensão, utiliza-se um circuito com amplificador operacional como inversor ou não inversor, pois o mesmo possui, naturalmente, alta impedância de entrada.

3.2 PROJETO PRÉ-AMPLIFICADOR

O circuito da figura 3.3, refere-se ao *Buffer* do estágio de entrada, agindo como um pré-amplificador com amplificador operacional na configuração não inversor.

Figura 3.3 – Pré-amplificador para condicionamento do sinal



Fonte: Produção do autor.

3.2.1 Circuito de polarização

Este circuito está representado no lado direito da figura 3.3, e também recebe o nome de circuito de *bias*. Como não haverá tensão de alimentação simétrica, faz-se a necessidade de um nível DC em todo o circuito analógico. O nível DC é calculado mediante o divisor resistivo entre os resistores R4 e R5. Calculado a seguir

$$V_{DC} = \frac{9V \times 10k\Omega}{10k\Omega + 10k\Omega} = 4,5V \quad (3.1)$$

Este valor é escolhido respeitando o fato de que deve-se haver a máxima excursão do sinal possível, sem que haja saturação dos AmpOs. O capacitor C4 é um filtro para eliminar a ondulação (*ripple*) da tensão, pois as fontes comerciais de 9V apresentam esse nível elevado. O valor de $10\mu F$ é arbitrado pelo projetista, mas pode variar entre $2,2\mu F$ à $100\mu F$. O capacitor C5 é um capacitor de desacoplamento, servindo com o intuito de eliminar ruídos da fonte de alimentação. Essas fontes são chaveadas, e naturalmente muito ruidosas, dessa forma, a tensão DC de saída deve ser filtrada. O valor de $100nF$ é praticamente adotado mundialmente em circuitos analógicos.

3.2.2 Impedância de entrada

Analisando a figura 3.3, o resistor R1 serve como impedância de entrada, uma vez que ele está em paralelo com a impedância de entrada do AmpOp Z_{in} . A equação a seguir calcula o valor da impedância de entrada.

$$Z_{in} = \frac{R_1 \times Z_{ampop}}{R_1 + Z_{ampop}} = \frac{10^6 \times 10^{12}}{10^6 + 10^{12}} \approx 10^6 \Omega \quad (3.2)$$

Como R1 é menor que Z_{in} , então a impedância de entrada pode ser considerada praticamente R1 (CLAYTON; WINDER, 2003). Além disso, esse resistor serve como resistor de polarização, adicionando um nível DC de 4,5V no sinal da guitarra. O capacitor C1 serve como capacitor de acoplamento, retirando níveis DC oriundos do captador através de um filtro passa-alta em conjunto com R1. A frequência de corte desse filtro inicial é calculada a seguir.

$$f_c = \frac{1}{2\pi R_1 C_1} = 1,59 \text{ Hz} \quad (3.3)$$

O valor de R1 está baseado em circuitos de pedais comerciais que apresentam impedância de entrada de $1M\Omega$. O capacitor C1 é definido pelo projetista. Deve-se salientar que a frequência de corte não deve exceder 20Hz , pois dessa forma, o circuito estaria influenciando a faixa audível.

3.2.3 Ganho do estágio de condicionamento

Os resistores R2, R3 e o potenciômetro dão a relação de ganho do circuito. A expressão do ganho desse circuito segue a mesma análise de um amplificador não inversor, e pode ser vista a seguir.

$$G = \left(1 + \frac{R_2}{R_{3pot} + R_4}\right) \quad (3.4)$$

A seguir encontra-se o cálculo deste ganho entre o valor mínimo e o máximo, com valores arbitrados de $22k\Omega$, $1k\Omega$ e $100k\Omega$ para R2, R3 e potenciômetro, respectivamente.

$$G_{min} = \left(1 + \frac{22k}{100k + 1k}\right) \approx 1,21 \rightarrow 1,71 \text{ dB} \quad (3.5)$$

$$G_{max} = \left(1 + \frac{22k}{0 + 1k}\right) = 22 \rightarrow 26,84 \text{ dB} \quad (3.6)$$

3.2.4 Análise em frequência do pré-amplificador

O pré-amplificador, além de ter uma frequência de corte de $1,59Hz$, possui outra frequência de corte em $7,234kHz$, dada pelo resistor $R2$ e capacitor $C2$, formando um filtro passa-baixa.

$$f_{c2} = \frac{1}{2\pi R2C2} = \frac{1}{2\pi 22k\Omega 1nF} = 7,234kHz \quad (3.7)$$

Circuitos comerciais utilizam esse filtro para "amenizar" a resposta em frequência do pré-amplificador, e prevenir instabilidades. Além disso, esse filtro ajuda a limitar a banda do sinal. Como será descrito posteriormente, deve-se ter um banda limitada para amostrar corretamente o sinal. A frequência de corte foi escolhida com base na banda de passagem dos captadores (ver figura 2.5).

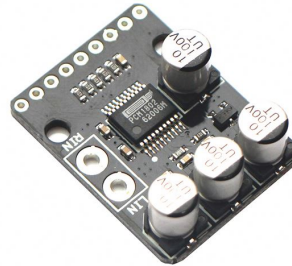
3.3 CONVERSOR AD

Para ser processado digitalmente, o sinal da guitarra deve passar por um conversor analógico digital (ADC). Esse conversor é responsável pela transformação de um sinal contínuo no tempo, em um sinal discreto, que nada mais é que uma sequência de números igualmente espaçados. Para realizar tal processo, sem que exista a perda de conteúdo, deve-se respeitar o teorema da amostragem (DINIZ; SILVA; NETTO, 2005).

Dessa forma, a escolha do conversor AD deve-se levar em conta algumas características. Primeiramente, precisa-se saber o número de bits do ADC. Para sinais de áudio, utiliza-se conversores de 24bits em aplicações de alto rendimento (KESTER, 2005). As frequências de amostragem acima de tudo, devem respeitar o teorema da amostragem, por padrão industrial, utiliza-se $44,1kHz$ para aplicações, como por exemplo, o CD. Em outras aplicações frequências maiores como $48kHz$ e $96kHz$ são mais utilizadas (KESTER, 2005). O aumento da frequência de amostragem prevê a diminuição do efeito chamado *aliasing*, que será discutido posteriormente.

Para este projeto, optou-se por utilizar o conversor AD PCM1802. O placa utilizada pode ser vista na figura a seguir. Na placa, o conversor já vem com capacitores de desacoplamento, capacitores para eliminar possíveis *ripples* de tensão e resistores limitadores de corrente nas linhas digitais.

Figura 3.4 – Conversor AD PCM1802



Fonte: (TECHNOLOGY, 2022)

Na tabela a seguir, encontram-se alguns parâmetros que justificam a escolha. Além disso, o fabricante recomenda a utilização do mesmo em aplicações como CD, instrumentos musicais elétricos e entre outras aplicações.

Tabela 3.1 – Parâmetros PCM1802

Descrição	Valor
Frequência de amostragem (f_s)	16kHz à 96kHz
Resolução (bits)	24
THD + N (dB)	96
Dynamic range ou DR (dB)	105
SNR (dB)	105
Protocolo de comunicação	I^2S
Tensão de alimentação	5V e 3.3V
Corrente	10mA para $f_s=48kHz$

Fonte: Adaptado de (INSTRUMENTS, 2016).

Como frequência de amostragem, adotou-se 96kHz por ser um padrão de aplicações de alto nível. Na tabela a seguir, encontra-se os pinos do conversor e sua conexão com o microcontrolador na coluna "configuração". Para este projeto, o conversor AD atuará no modo "*slave*". Nesse modo, todos os *clocks* necessários são providos pelo microcontrolador.

Os pinos do conversor que não possuem conexão com o microcontrolador, estão configurados via *hardware* como nível lógico alto ou baixo, pois o sistema embarcado não terá outras configurações. A escolha dos níveis lógicos, são escolhidos com base na folha de dados do componente.

Tabela 3.2 – Conexões do conversor AD

Pino	Descrição	Configuração
<i>VinR</i>	Entrada analógica canal direito	Analógico
<i>VinL</i>	Entrada analógica canal esquerdo	Analógico
BYPASS	Ativa filtro interno (passa alta) e tira nível DC na saída	Baixo
FSYNC	Ativa a transmissão de dados entre conversor AD e o microcontrolador	Alto
LRCK	<i>Clock</i> na frequência de amostragem	PB10
MODE1	Seleciona o modo (" <i>slave</i> " para este projeto)	Baixo
MODE0	Seleciona o modo (" <i>slave</i> " para este projeto)	Baixo
FMT1	Seleciona o formato (" <i>I²S</i> " para este projeto)	Baixo
FMT0	Seleciona o formato (" <i>I²S</i> " para este projeto)	Alto
OSR	Frequência de amostragem	PB10
SCKL	<i>Clock</i> do sistema	PC6
DOUT	Saída de dados amostrados	PC3

Fonte: Adaptado de (INSTRUMENTS, 2016).

3.4 FILTRO ANTI *ALIASING* - TEORIA

Nesta seção será discutido a importância da filtragem dos sinais de entrada, assim como o projeto de um filtro *anti-aliasing* para o pedal.

Esse tipo de filtro é necessário em qualquer sistema de aquisição de dados, onde trabalha-se com amostragem de sinais.

Um sinal amostrado pode ser descrito como o sinal analógico multiplicado por um trem de impulsos espaçados pela frequência de amostragem. A expressão matemática é dada a seguir.

$$x_i(t) = x_a(t) \times \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \quad (3.8)$$

Onde x_a é o sinal analógico, que representa o sinal da guitarra. A transformada de Fourier da expressão anterior é:

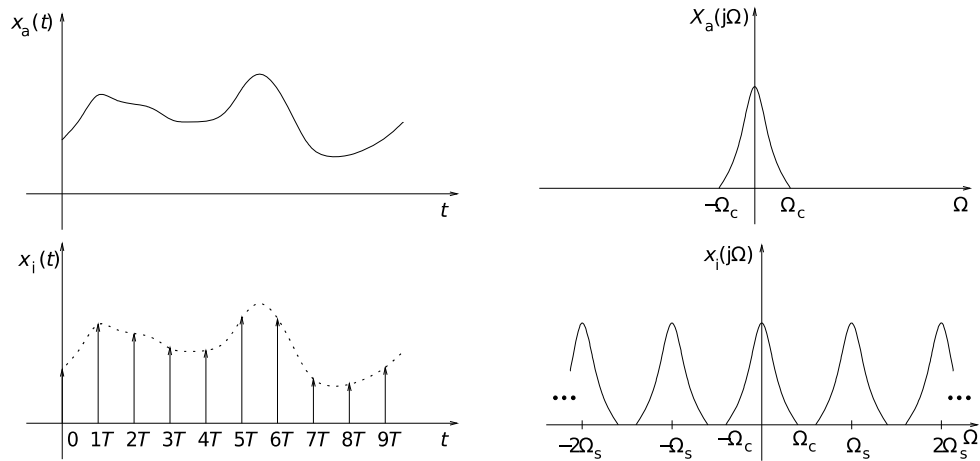
$$X_i(j\Omega) = \frac{1}{2\pi} X_a(j\Omega) * P(j\Omega) \quad (3.9)$$

Onde $P(j\Omega)$ é a transformada de Fourier da função impulso. Reajustando a equação anterior, tem-se:

$$X_i(j\Omega) = \frac{1}{T_s} X_a(j\Omega) * \sum_{k=-\infty}^{\infty} \delta\left(\Omega - \frac{2\pi k}{T_s}\right) \quad (3.10)$$

A convolução na equação (3.10), entre uma função e um impulso deslocado, é a própria função deslocada no domínio da frequência. A figura a seguir demonstra um sinal analógico e o sinal amostrado, assim como os espectros de cada um.

Figura 3.5 – Espectros dos sinais



Fonte: (DINIZ; SILVA; NETTO, 2005)

Observando a figura anterior, fica evidente que: para evitar que as cópias repetidas no espectro se interfiram uma nas outras, o sinal deve ter uma banda limitada. Além disso, esse fenômeno representa o teorema de Nyquist. Esse teorema diz que: se um sinal possuir banda limitada, então, o mesmo pode ser completamente reconstruído se a frequência de amostragem (na figura Ω_s) for duas vezes maior que a maior frequência contida no sinal ($\Omega_s = 2\Omega_c$).

Dessa forma, deve-se limitar a banda do sinal para que todas as frequências menores que $\frac{f_s}{2}$ sejam fielmente amostradas. Tal processo é construído aplicando um filtro do tipo passa-baixa no sinal. Com o surgimento de técnicas de processamento digital de sinais, podem-se obter filtros digitais com uma resposta bem parecida ao ideal, porém, esses filtros estariam filtrando o sinal com ruídos. Dessa forma, filtros analógicos ainda possuem espaço nessa questão, pois atuam antes do valor ser convertido para o conversor AD (BAKER, 1999).

Os filtros, além de serem caracterizados pelo seu tipo, passa-baixa, passa-alta e entre outros, podem ser caracterizados pela sua resposta em frequência. Os mais comuns são: *Butterworth*, *Bessel* e *Chebyshev*. O filtro *Butterworth* se destaca por ter uma resposta em frequência plana na banda de passagem (melhor do que os outros filtros). Portanto, a resposta em frequência para o filtro deste projeto será do tipo *Butterworth*.

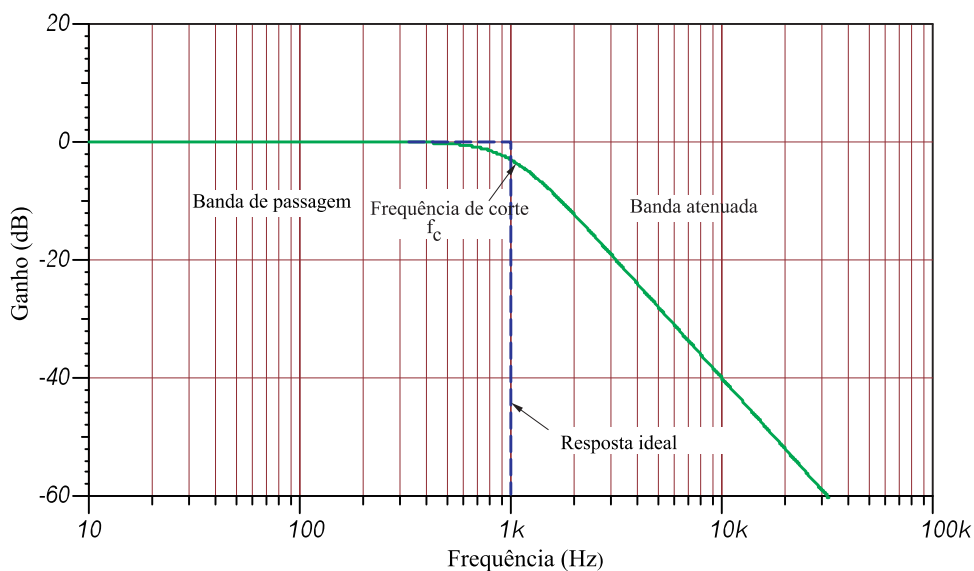
Entre as topologias de filtros ativos, destaca-se a *Sallen-Key*. Este filtro apresenta estágios com ganho unitário (o filtro não necessita dar ganho ao circuito, porque a etapa de condicionamento de sinal já o faz), e atua em baixas frequências (20Hz à 20kHz). Em resumo, o filtro deste projeto será do tipo *Butterworth* e terá a topologia *Sallen-Key*.

Existem cinco parâmetros importantes para o projeto, são eles: atenuação máxima na banda de passagem (α_{max}), atenuação mínima ao final da banda de transição (α_{min}), frequência de corte (f_c), frequência que encerra a banda de transição (f_{stop}) e a ordem do filtro (N).

Com o auxílio da figura a seguir, entende-se como banda de passagem toda a faixa de frequência que terá ganho unitário ou superior, até a frequência de corte. A banda de atenuação consiste em duas partes: banda de transição e banda de parada. A banda de transição é inerente aos filtros reais. Da figura 3.6, pode-se observar o comportamento de um filtro com resposta ideal. Esse comportamento é inviável, pois requer filtros de ordem muito superior, o que implica no uso de vários AmpOps. Dessa forma, para filtros reais é comum o uso de uma banda de transição que consiste na faixa de frequência entre a frequência de corte e metade da frequência de amostragem ($\frac{1}{2}f_s$). A banda de transição ficará cada vez menor a medida que a ordem do filtro aumenta.

A banda de parada consiste na faixa de frequência onde a atenuação desejada foi atingida.

Figura 3.6 – Resposta em frequência do filtro *Butterworth*



Fonte: Produção do próprio autor

A tabela a seguir demonstra os dados selecionados.

Tabela 3.3 – Resumo dos parâmetros filtro *anti-aliasing*

Descrição	Valor
Frequência de amostragem (f_s)	96kHz
Frequência de corte (f_c)	8kHz
Frequência da banda de parada (f_{stop})	48kHz
Atenuação na banda de passagem (A_p)	1dB
Ordem do filtro (N)	5

Fonte: Produção do próprio autor.

A banda de passagem utiliza uma frequência de corte do filtro (f_c) de 8kHz (com base na figura 2.5). Para a banda de parada, (f_{stop}), adotou-se a frequência de 48kHz, que é a metade da frequência de amostragem (f_s).

A equação a seguir demonstra como determinar a ordem de um filtro com resposta do tipo *Butterworth* (SCHAUMANN; VALKENBURG, 2001).

$$N = \frac{\log \left[\frac{[10^{0,1\alpha_{min}} - 1]}{[10^{0,1\alpha_{max}} - 1]} \right]}{2 \log \frac{\omega_s}{\omega_p}} \quad (3.11)$$

A atenuação mínima (α_{min}), representa o valor em dB do quanto o sinal deve ser atenuado na frequência de parada (f_{stop}). Tal atenuação pode ser calculada como sendo a relação sinal-ruído (SNR) do sistema (KESTER, 2008). Para um conversor AD de 24 bits tem-se:

$$SNR = 6,02 \times 24 + 1,76 = 145,76 \text{ dB} \quad (3.12)$$

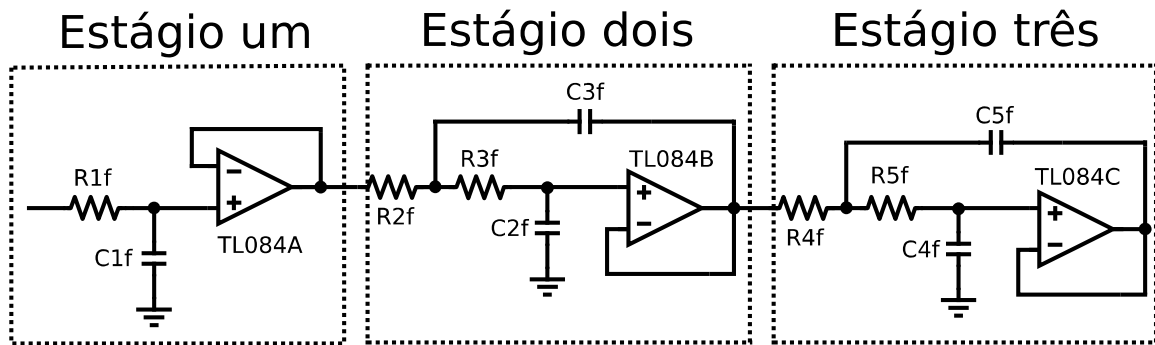
A atenuação máxima na banda de passagem é um valor arbitrário definido pelo projetista. Para este projeto o valor foi escolhido conforme a tabela 3.3. Com a frequência ω_p sendo a frequência de corte e ω_s a frequência de parada, ambas em *rad/s*, tem-se a ordem do filtro como:

$$n = \frac{\log \left[\frac{(10^{0,1 \times 146,24} - 1)}{(10^{0,1 \times 1} - 1)} \right]}{2 \log \left[\frac{48k}{8k} \right]} \approx 10 \quad (3.13)$$

De acordo com a equação anterior, é inviável projetar um filtro com essa ordem de forma analógica, pois acarretaria em vários amplificadores operacionais. Desse modo, optou-se por diminuir drasticamente o ordem do filtro. Adotou-se então um filtro de 5ª ordem. Tal escolha é baseada em trabalhos anteriores como Lang (2018), Alves (2018) e na seção de resultados será demonstrado que a ordem do mesmo é suficiente.

A figura a seguir demonstra um filtro passa-baixa, com topologia *Sallen-Key*, resposta do tipo *Butterworth* e ordem cinco.

O filtro consiste de três estágios. O primeiro estágio é um filtro passa baixa ativo de primeira ordem. Na sequência existem dois filtros ativos de segunda ordem com polos complexos, que completam a ordem cinco do filtro. A atenuação total do filtro pode ser calculada como $20n$, onde "n" é a ordem do filtro. Dessa forma, tem-se -100dB de ganho ao invés dos $-146,24\text{dB}$.

Figura 3.7 – Projeto do filtro *Butterworth* 5ª Ordem

Fonte: Produção do próprio autor

A seguir tem-se uma tabela com os valores escolhidos de cada componente passivo e a análise de cada estágio.

Tabela 3.4 – Valores dos componentes do filtro

Componente	Valor calculado	Valor adotado
R1f	1,99k Ω	2k Ω
C1f	10nF	10nF
R2f	1,61k Ω	1,6k Ω
R3f	1,61k Ω	1,6k Ω
C2f	10nF	10nF
C2f	15,28nF	10nF
R4f	614,6 Ω	680 Ω
R5f	614,6 Ω	680 Ω
C4f	10nF	10nF
C5f	104,77nF	100nF

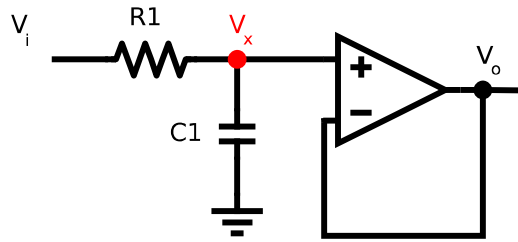
Fonte: Produção do próprio autor.

Os valores da tabela foram calculados utilizando o programa *Filter Pro* da *Texas Instruments*. Basicamente, o projetista escolhe o tipo de resposta (*Butterworth*, *Bessel* e *etc*), a configuração (*Sallen-Key*), se é passa-baixa ou outro e insere os dados como frequência de corte e ordem do filtro. Com isso, o programa apresenta valores para os componentes passivos (resistores e capacitores). Nas próximas seções será analisado se os valores comerciais adotados batem com a definição inicial do projeto.

3.4.1 Estágio um

O primeiro estágio do filtro anti-alias consiste em um filtro passa-baixa de primeira ordem. A figura a seguir mostra o circuito.

Figura 3.8 – Análise do primeiro estágio



Fonte: Produção do próprio autor

O equacionamento de tal filtro pode ser encontrado a seguir. Analisando o nó V_x , tem-se as equações a seguir.

$$\frac{V_x}{\frac{1}{sC_1}} + \frac{V_x - V_i}{R_1} = 0 \quad (3.14)$$

$$V_x = V_o \quad (3.15)$$

Substituindo (3.15) em (3.14), e reajustando os termos da equação, pode-se encontrar:

$$\frac{V_o}{V_i} = \frac{1}{sC_1R_1 + 1} \quad (3.16)$$

Evidenciando a ordem do filtro e a frequência de corte. A frequência de corte (o polo da equação anterior), pode ser calculada como:

$$f_c = \frac{1}{2\pi R_1 C_1} = \frac{1}{2\pi 2k\Omega 10nF} = 7,95kHz \quad (3.17)$$

3.4.2 Estágio dois

Primeiramente, será feito uma análise da função de transferência de um filtro passa baixa de segunda ordem e depois será realizado o equacionamento adequado para este projeto. A função de transferência deste estágio é vista a seguir.

$$T(s) = \frac{\pm H \omega_o^2}{s^2 + \frac{\omega_o}{Q} + \omega_o^2} \quad (3.18)$$

Onde ω_o é a frequência de corte desejada, H é o ganho da função de transferência e Q é o fato de qualidade.

Pode-se dividir s por ω_o para obter uma frequência normalizada chamada de $s_n = \frac{s}{\omega_o}$

$$T(s) = \frac{\pm H}{\left(\frac{s}{\omega_o}\right)^2 + \frac{s}{Q\omega_o} + 1} = \frac{\pm H}{s_n^2 + \frac{1}{Q}s_n + 1} \quad (3.19)$$

Substituindo agora $s = j\omega$ e adicionando um fator de normalização (FSF), pode-se obter a equação a seguir.

$$T(f) = -\frac{H}{\left(\frac{f}{FSF f_c}\right)^2 + \frac{1}{Q} \frac{jf}{FSF f_c} + 1} \quad (3.20)$$

Da equação anterior, existem três regiões: abaixo da frequência de corte, na frequência de corte, e acima da frequência de corte.

- Quando $f < f_c$: $T(f) = H$, e o circuito passa os sinais com um ganho H.
- Quando $\frac{f}{f_c} = FSF$, $T(f) = -jHQ$ os sinais têm sua fase alterada para 90° e sua amplitude é atenuada com o fator de qualidade Q
- Quando $f > f_c$, $T(f) = -H\left(\frac{FSF f_c}{f}\right)^2$, os sinais têm sua fase alterada para 180° e sua amplitude atenuada pelo quadrado da frequência.

Essas três regiões analisadas indicam claramente o comportamento de um filtro passa-baixas. A figura a seguir demonstra uma tabela com valores padronizados dos parâmetros FSF e fator de qualidade (Q). Os valores são para uma resposta do tipo *Butterworth*. Tal tabela é importante, pois como essas topologias já estão bem consolidadas, a mesma auxilia no desenvolvimento do projeto.

Figura 3.9 – Análise do primeiro estágio

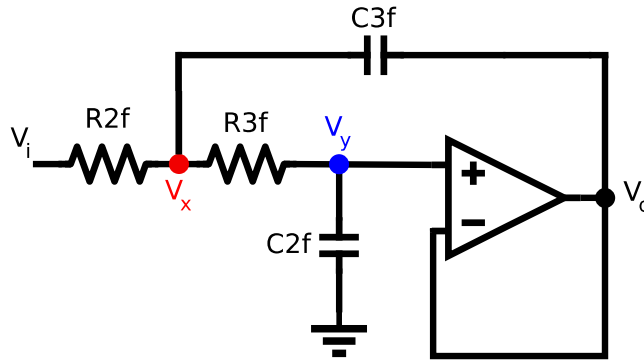
Ordem	Estágio 1		Estágio 2		Estágio 3		Estágio 4		Estágio 5	
	FSF	Q	FSF	Q	FSF	Q	FSF	Q	FSF	Q
2	1.000	0.7071								
3	1.000	1.0000	1.000							
4	1.000	0.5412	1.000	1.3065						
5	1.000	0.6180	1.000	1.6181	1.000					
6	1.000	0.5177	1.000	0.7071	1.000	1.9320				
7	1.000	0.5549	1.000	0.8019	1.000	2.2472	1.000			
8	1.000	0.5098	1.000	0.6013	1.000	0.8999	1.000	2.5628		
9	1.000	0.5321	1.000	0.6527	1.000	1.0000	1.000	2.8802	1.000	
10	1.000	0.5062	1.000	0.5612	1.000	0.7071	1.000	1.1013	1.000	3.1969

Fonte: Adaptado de Karki (2002).

Vale ressaltar que a coluna "estágio 3" da figura 3.9 refere-se ao filtro de primeira ordem, pois não há a existência do fator de qualidade para ordem um. A coluna "estágio 1" refere-se ao estágio dois da figura 3.7 (primeiro filtro de segunda ordem) e, por fim, a coluna "estágio 2" é referente ao estágio três da mesma figura.

Na figura a seguir pode-se encontrar a topologia *Sallen-Key* para um filtro passa baixa ativo de segunda ordem.

Figura 3.10 – Análise filtro de segunda ordem *Sallen -Key*



Fonte: Produção do próprio autor

Através da análise nodal nos nós V_x e V_y , pode-se obter as equações a seguir.

$$\frac{V_x - V_i}{R2_f} + \frac{V_x - V_o}{\frac{1}{sC3_f}} + \frac{V_x - V_y}{R3_f} = 0 \quad (3.21)$$

$$\frac{V_y - V_x}{R3_f} + \frac{V_y}{\frac{1}{sC2_f}} = 0 \quad (3.22)$$

$$V_y = V_o \quad (3.23)$$

Para encontrar a função de transferência, deve-se substituir (3.23) em (3.22) (equação (3.24)). Com isso, pode-se obter uma expressão resultante, relacionando o nó V_x com V_o . Por fim, pode-se isolar a tensão de entrada e saída na equação (3.21). A equação final pode ser vista na equação (3.25) (onde foram omitidas as etapas de manipulação algébrica).

$$\frac{V_o - V_x}{R3_f} + sC2_f V_o = 0 \rightarrow V_x = (1 + R3_f sC2_f) V_o \quad (3.24)$$

$$\frac{V_o}{V_i} = \frac{1}{s^2 C2_f C3_f R2_f R3_f + sC2_f (R2_f + R3_f) + 1} \quad (3.25)$$

A equação final é reajustada como a seguir.

$$\frac{V_o}{V_i} = \frac{\frac{1}{C2_f C3_f R2_f R3_f}}{s^2 + s \left[\frac{1}{C3_f} \frac{1}{R3_f + R2_f} \right] + \frac{1}{C2_f C3_f R2_f R3_f}} \quad (3.26)$$

A frequência de corte é calculada a seguir.

$$f_c = \frac{1}{2\pi\sqrt{R2_f R3_f C2_f C3_f}} = \frac{1}{2\pi\sqrt{1,6k\Omega 1,6k\Omega 10nF 10nF}} = 9,9522kHz \quad (3.27)$$

O fator de qualidade é calculado a seguir.

$$Q = \frac{1}{2\pi f_c C2_f (R2_f + R3_f)} = 0,5 \quad (3.28)$$

A diferença entre os valores calculados e adotados reflete na diferença da frequência de corte e fator de qualidade. Entretanto, os valores adotados mostraram um bom resultado prático.

3.4.3 Estágio três

Para o estágio três, a análise é a mesma. Porém, o fator de qualidade é alterado. Com o auxílio do *FilterPro* (comentando anteriormente), tem-se o cálculos dos componentes passivos. Nesta seção será apenas confirmado se a frequência de corte, e o fator de qualidade se adequam com o projeto.

A frequência de corte para o terceiro estágio é dada a seguir.

$$f_c = \frac{1}{2\pi\sqrt{680\Omega 680\Omega 10nF 100nF}} = 7,401kHz \quad (3.29)$$

O fator de qualidade é calculado a seguir.

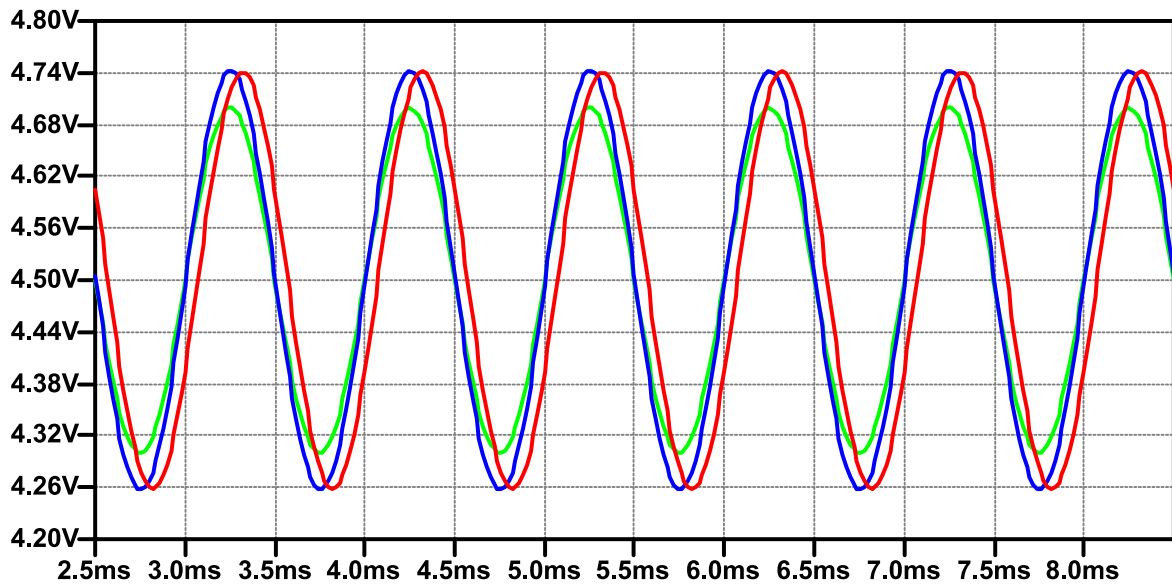
$$Q = \frac{1}{2\pi f_c C4_f (R4_f + R5_f)} = 1,58 \quad (3.30)$$

Novamete, os resultados anteriores mostraram um bom resultado experimental.

3.5 FILTRO ANTIALIASING - SIMULAÇÃO

Nesta seção será apresentado dados de simulação utilizando o programa *LTspice XVII*. Para a primeira simulação, pode-se observar o sinal de entrada no conversor AD, que passa pelo pré-amplificador e o filtro. Para o sinal de entrada utiliza-se um senóide com frequência de $1kHz$. O sinal em verde representa a senóide de entrada com um nível DC de $4,5V$. A onda em azul representa a tensão após a saída do estágio do pré-amplificador, e a tensão em vermelho representa a tensão após a passagem do filtro passa-baixa.

Figura 3.11 – Sinal no domínio do tempo para o filtro anti *aliasing*

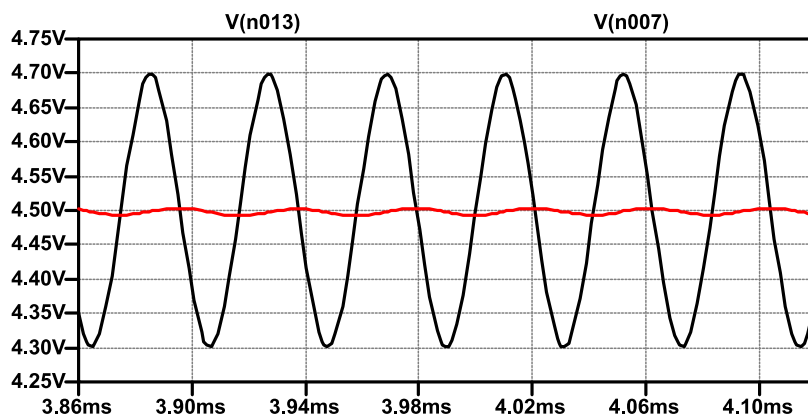


Fonte: Produção do próprio autor

Da figura anterior, percebe-se que os circuitos propostos atendem a necessidade do projeto. O atraso entre as tensões na saída do pré-amplificador e na saída do filtro é de $69,10\mu s$. Percebe-se que entre o sinal de entrada (em verde) e o sinal após o amplificador existe um ganho de tensão de aproximadamente 1V, como projetado. Outro aspecto importante, o sinal após o filtro não tem sua amplitude modificada, desde que, esteja dentro da banda de frequência especificada.

Simulando uma frequência de $24kHz$ de entrada, pode-se observar uma atenuação considerável na tensão de saída (sinal em vermelho) na figura 3.12.

Figura 3.12 – Tensão após o filtro *antialiasing* com senoide de entrada de $24kHz$



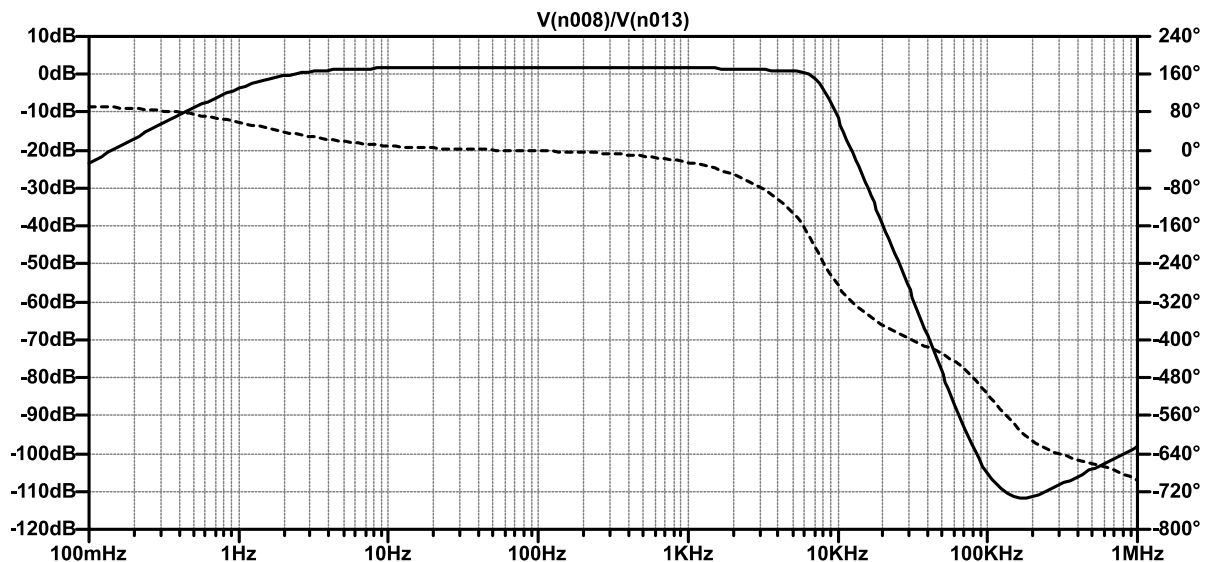
Fonte: Produção do próprio autor

A figura anterior demonstra a grande atenuação imposta pelos filtros. A senoide em preto demonstra a tensão de entrada, com $24kHz$ e tensão de $200mV$ de pico, com um nível DC de

4,5V. A onda em vermelho demonstra a tensão de saída do filtro. A tensão máxima de saída foi medida como 4,503V, praticamente eliminando todo o sinal injetado.

A figura a seguir demonstra a resposta em frequência do pré-amplificador em conjunto com o filtro.

Figura 3.13 – Resposta em frequência apenas do fitro



Fonte: Produção do autor

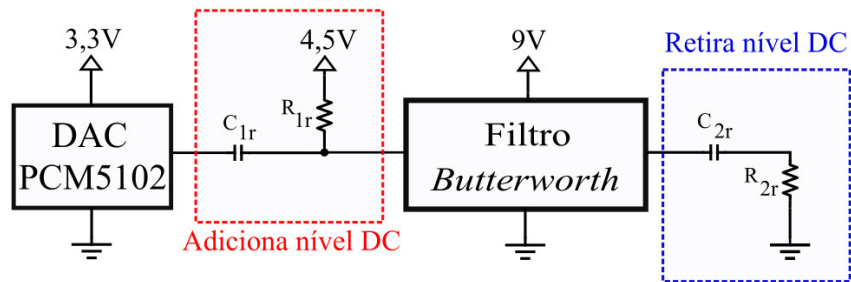
Com frequência de corte simulada de $7,58kHz$, e atenuação em $48kHz$ de $-77dB$, pode-se perceber a característica passa-banda do sistema, o ganho constante na banda de passagem e uma transição abrupta na banda de transição. A simulação foi realizada considerando os valores comerciais adotados com variação paramétrica de 5% para os componentes passivos.

3.6 FILTRO DE RECONSTRUÇÃO

O filtro de reconstrução é, essencialmente, o mesmo que o filtro de entrada. Deve-se ter a mesma banda de passagem, preferencialmente não atenuar ou inserir ganho no sinal, e ter uma baixa impedância de saída. Este filtro é posicionado logo após o conversor DA, e limita a banda de passagem do sinal novamente, eliminando o conteúdo de alta frequência, que é inserido através do processador de conversão digital-analógico. A figura a seguir apresenta os circuitos adicionais necessários para o filtro de reconstrução.

Antes do filtro, é adicionado um nível DC ao sinal, pois a saída do filtro possui tensão negativa. O capacitor C_{1r} acopla o sinal do DAC com o filtro, e o resistor R_{1r} adiciona a polarização ao sinal.

Figura 3.14 – Circuito para o filtro de reconstrução



Fonte: Produção do autor

Além disso, na saída deste filtro é adicionado um circuito RC do tipo passa-alta. A frequência de corte do filtro RC de saída é:

$$f_c = \frac{1}{2\pi * RC} = \frac{1}{2\pi * 1k10\mu} = 15,91Hz \quad (3.31)$$

O resistor R_{2r} ($1k\Omega$) também é responsável pela impedância de saída do sistema.

A tabela a seguir apresenta os valores adotados.

Tabela 3.5 – Valores dos componentes do filtro

Componente	Valor
R_{1r}	$1M\Omega$
C_{1r}	$100nF$
R_{2r}	$1k\Omega$
C_{2r}	$1\mu F$

Fonte: Produção do autor.

3.7 CONVERSOR DA

O conversor DA (DAC) escolhido foi o PCM5102 da *Texas Instruments*. A escolha desse componente se baseia na praticidade de controle da conversão, que é a mesma que o conversor AD. A frequência de amostragem do componente se estende até 392kHz. As resoluções são de 16 a 32bits. O conversor é do tipo *Delta-Sigma*. As aplicações do mesmo, segundo o fabricante, orbita em sistema de áudio, DVD *players* e HD/TV.

A figura a seguir demonstra os pinos do componente.

Figura 3.15 – Pinos conversor DA



Fonte: (INSTRUMENTS, 2012)

A tabela a seguir descreve as conexões do DAC com o microcontrolador. Assim como o ADC, o conversor DA será o escravo e o microcontrolador o mestre, provendo todos os *clocks* necessários. A configuração do conversor é realizada através das entradas digitais do circuito integrado (CI). Foi escolhido o formato de transmissão como I2S, 24 *bits*, sem *soft mute* e latência normal.

Tabela 3.6 – Conexões ADPCM1802

Pino	Descrição	Nível lógico
<i>OutR</i>	Saída analógica canal direito	<i>Jack</i> de saída
<i>OutL</i>	Saída analógica canal esquerdo	<i>Jack</i> de saída
DEMP	<i>De-emphasis</i> para 44,1kHz	Baixo
XSMT	<i>Soft mute</i>	Alto
FMT	Formato de áudio	Baixo
LRCK	<i>Clock</i> na frequência de amostragem	PB12
DIN	Entrada de áudio	PC3
BCK	Frequência de amostragem	PB10
SCKL	<i>Clock</i> do sistema	PC6
FLT	Seleção de filtro	Baixo

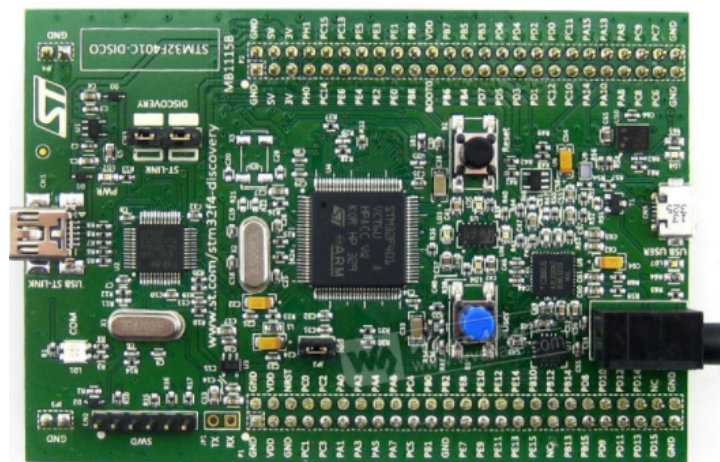
Fonte: (INSTRUMENTS, 2012).

Os demais pinos estão relacionados com a filtragem da fonte de alimentação e estão conectados a capacitores de desacoplamento. Como para este projeto foi adquirido um módulo pronto, tem-se acesso somente aos pinos listados na tabela anterior, e não há necessidade de utilizar os pinos restantes.

3.8 MICROCONTROLADOR

Esta seção dedica-se a escolha do microcontrolador, e apresentará alguns parâmetros do mesmo. Para este projeto foi selecionado o microcontrolador (MCU) STM32F407 *Discovery* da *ST Microelectronics*, este MCU pode ser observado na figura a seguir, onde é apresentado a placa contendo o circuito integrado STM32F407VGT6. A placa contém todos os circuitos necessários para o funcionamento do microcontrolador, como capacitores de desacoplamento, oscilador externo e também permite o *debug* via USB.

Figura 3.16 – Placa escolhida



Fonte: (STMICROELECTRONICS, 2021)

A tabela a seguir mostra os periféricos utilizados, assim como alguns dados do microcontrolador.

Tabela 3.7 – Periféricos do microcontrolador

Periférico	Descrição
<i>Clock</i>	168MHz
<i>Flash</i>	1MB
<i>RAM</i>	192kB
<i>I²S</i>	<i>Full-duplex 16 bits</i>
Controlador DMA	Possui dois com 8 canais em cada
ADC	12 <i>bits</i> height

Fonte: (STMICROELECTRONICS, 2021).

Uma das principais contribuições deste projeto é trabalhar com processamento de áudio em tempo real sem a utilização de um processador de sinais (DSP).

Segundo Orfanidis (2010), um DSP possui um multiplicador e acumulador (as duas operações em uma instrução) para efetuar o trabalho exaustivo dos algoritmos de processamento.

Este MCU, conta com instruções MAC e SIMD (performa várias instruções MAC em um ciclo de *clock*). Além disso, o microcontrolador conta com FPU (*floating point unit*) e acesso direto da memória (DMA).

A família de microcontroladores córtex-M4 foi criada com um propósito de adicionar capacidades para algoritmos de processamento de sinais, tendo em vista que a linha anterior (M3) não possuía.

Além disso, o custo desta placa é menor em comparações com placas de desenvolvimento envolvendo processadores de sinais. Sendo assim, pode-se escolher este microcontrolador e atender os requisitos para processar o áudio em tempo real com boa qualidade.

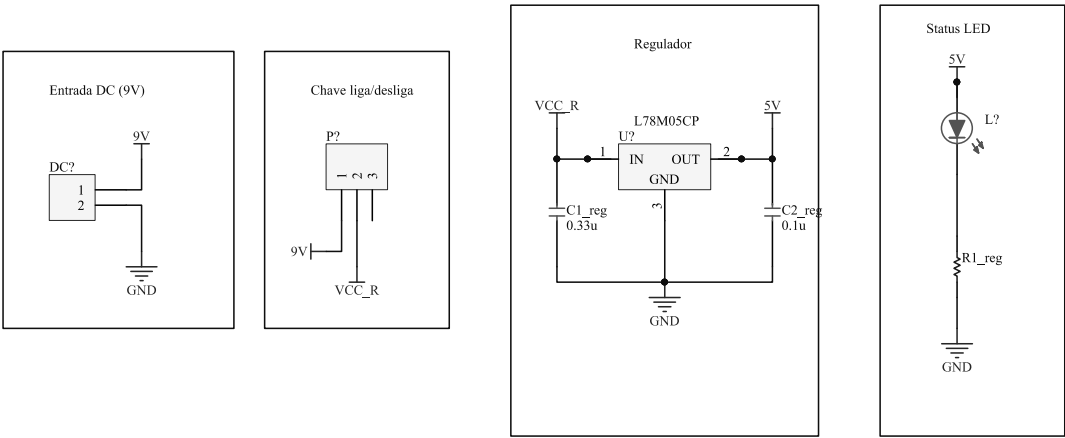
3.9 REGULADOR DE TENSÃO

O pedal de guitarra será alimentado por uma fonte de 9V DC, pois é um padrão comercial. No entanto, o microcontrolador e os conversores utilizados possuem tensão de alimentação diferente de 9V, o que se faz necessário o uso de um regulador. Para este projeto, foi selecionado o regulador LM7805. O principal motivo é a facilidade de projeto e a capacidade de suprir a corrente necessária, além da robustez do mesmo, que contempla proteção de curto-circuito e limitador de corrente. A tensão de entrada do regulador pode ser de até 25V, com tensão de saída fixa em 5V. A corrente de saída máxima é de 1,5A, o que é muito superior a corrente necessária.

Dessa forma, o microcontrolador será alimentado com 5V (proveniente do regulador) e os conversores AD e DA serão conectados ao pino de 3,3V do próprio MCU. Os AmpOps, por sua vez, serão conectados em 9V.

A figura a seguir demonstra a etapa de fonte de alimentação, envolvendo o regulador do projeto. Inicialmente, tem-se um conector que recebe a tensão de 9V DC. Esse conector é ligado a uma chave de três terminais para servir como liga/desliga. Em seguida, tem-se o regulador com os capacitores de filtragens. Os valores são de $330nF$ para a entrada e $100nF$ para a saída. Os mesmos foram retirados da folha de dados do componente. Como circuito auxiliar, foi acrescentado um LED para indicar que o circuito está ligado. O resistor R_{1reg} é o resistor limitador de corrente do LED, e é ajustado de forma experimental (cada projetista define a quantidade de brilho).

Figura 3.17 – Circuitos de alimentação



Fonte: Produção do autor.

3.10 AMPLIFICADORES OPERACIONAIS

Utilizando o programa *FilterPro*, ao calcular o filtro *antialiasing*, obtém-se um recurso que mostra a banda mínima dos amplificadores operacionais necessária para utilizar nos filtros. As mesmas são mostradas a seguir.

Tabela 3.8 – Banda mínima para os AmpOps

Estágio	Banda de passagem
1	400kHz
2 ($Q \leq 1$)	496kHz
3 ($Q > 1$)	1,296MHz

Fonte: Produção do autor.

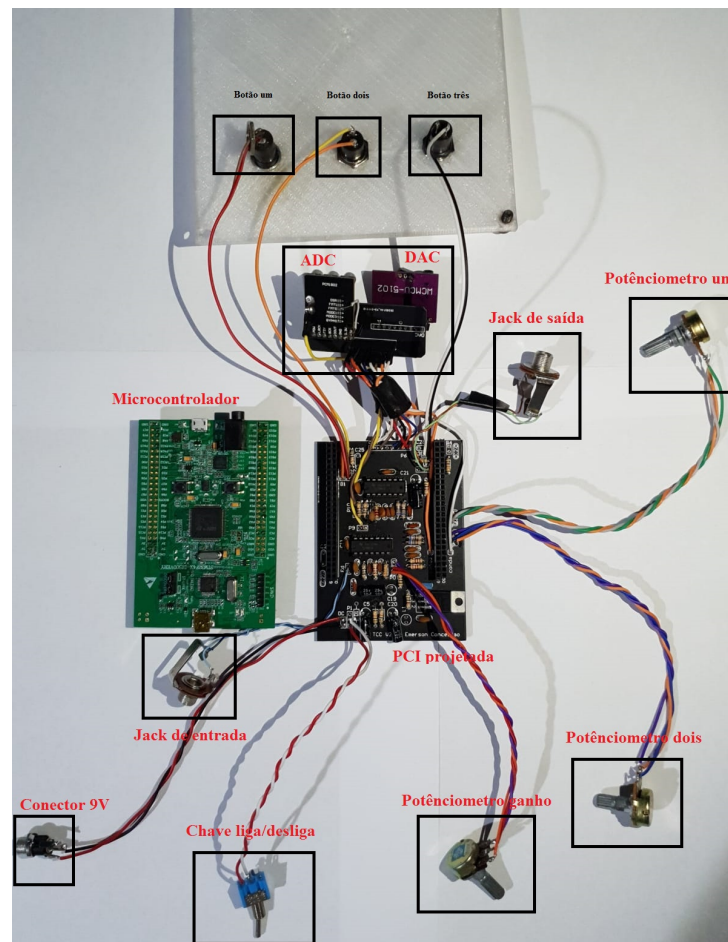
Como amplificador operacional para os circuitos e filtros, optou-se pelo TL084. Este amplificador possui tensão de alimentação de até 36V, proteção contra curto-circuito, alto *slew rate* ($16V/\mu s$) e banda de operação de até 4MHz.

Além disso, cada CI possui quatro amplificadores embarcados, diminuindo a complexidade no projeto da placa de circuito impresso. A tensão de alimentação de cada AmpOp será de 9V (com a chave da figura 3.17 ligada).

3.11 SÍNTESE DO CAPÍTULO

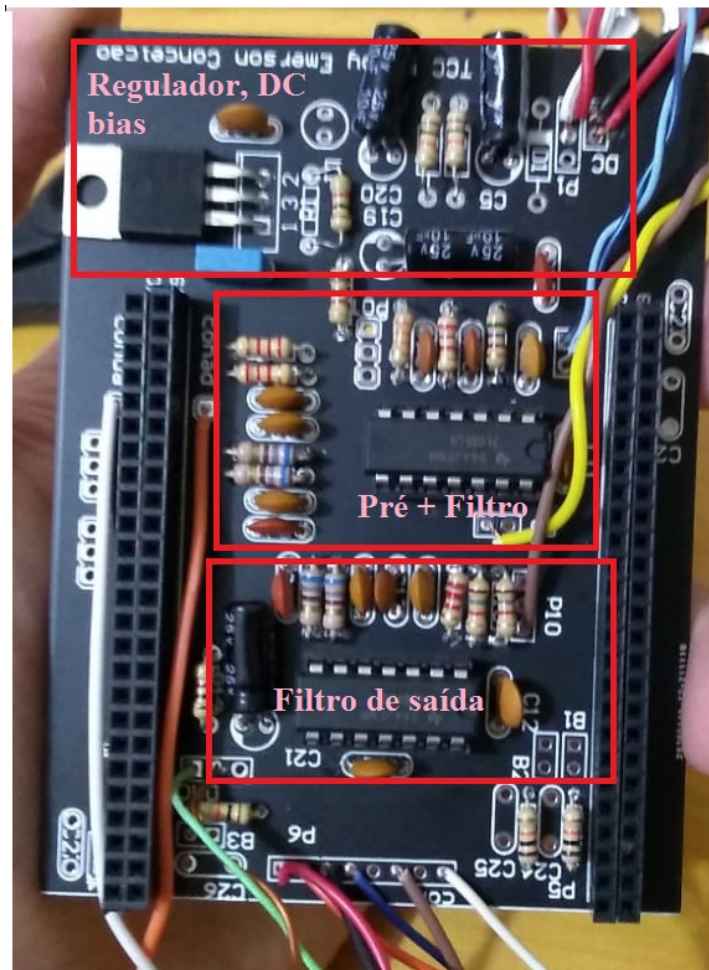
Este capítulo procurou tratar de todos circuitos utilizados, sua descrição e funcionamento. A síntese do mesmo consiste em duas placas de circuito impresso mostradas nas figuras que se seguem. A primeira contém todos os filtros e circuitos projetados e a segunda, apenas conectores para o conversor ADC e DAC. Foi escolhido separar os conversores do microcontrolador, com o intuito de separar os circuitos digitais dos analógicos. A placa dos conversores consiste apenas em conectores e não será demonstrada. A placa contendo os filtros pode ser vista melhor na figura 3.19.

Figura 3.18 – Sistema projetado



Fonte: Produção do autor.

Figura 3.19 – Placa de circuito impresso analógica



Fonte: Produção do autor.

Na parte superior da placa pode-se observar o regulador, com os capacitores de filtragem, assim como o circuito de bias, as conexões com a chave e o conector da fonte. Logo abaixo, encontra-se o primeiro TL084 responsável pelo amplificador operacional do pré-amplificador e filtro de entrada. Em seguida, observa-se o TL084 responsável pelo filtro de saída. Por fim, na parte inferior da placa, pode-se encontrar as conexões com a placa dos conversores.

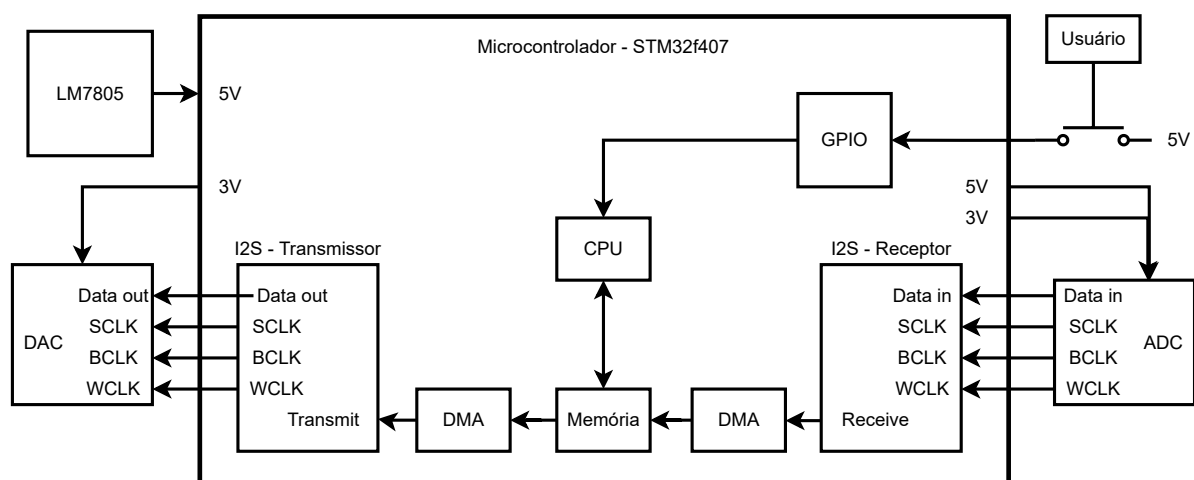
O projeto consta ainda com dois potenciômetros que serão manipulados pelo usuário para alterar determinados parâmetros nos efeitos, os mesmos são demonstrados na figura 3.18. A conexão dos mesmos ao MCU se dá pelos pinos PA3 e PA5.

4 SOFTWARE PARA O PROJETO

Neste capítulo será abordado o projeto sob o ponto de vista do *firmware*, ou seja, do código interno do pedal. Esse código o usuário não terá acesso. As variáveis que o usuário terá acesso, serão comentadas e evidenciadas.

A figura a seguir demonstra a arquitetura interna do projeto.

Figura 4.1 – Arquitetura interna do pedal



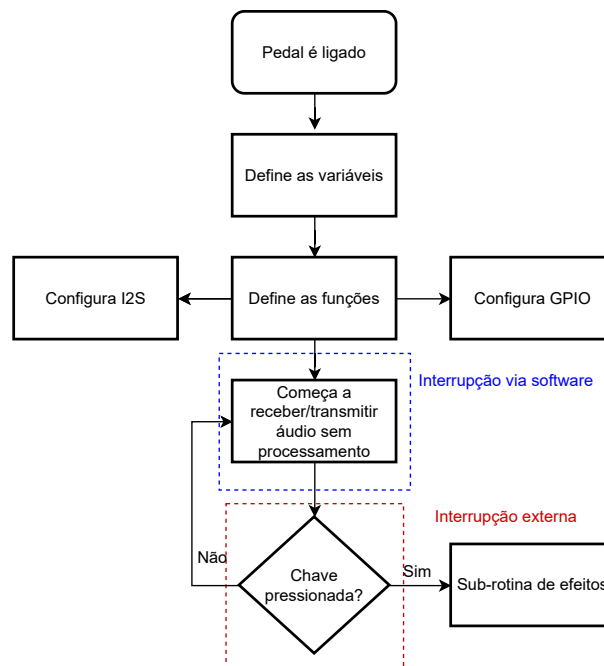
Fonte: Produção do próprio autor

Como componente principal destaca-se o microcontrolador que deverá ser capaz de armazenar todas as amostras de áudio por um período de tempo definido pelo usuário do pedal, assim como, realizar todos os cálculos necessários dentro de um intervalo sem que haja atrasos.

Dentro do microcontrolador deverá ser configurado alguns periféricos para que o áudio seja corretamente captado e transmitido, são eles: o periférico SPI/I2S, DMA, os periféricos responsáveis por comandar os pinos de entrada e saída (GPIO) e o conversor AD interno de *12bits* (omitido da figura anterior).

4.1 CÓDIGO PRINCIPAL

A figura a seguir mostra o fluxograma carregando a ideia principal para o desenvolvimento do código para o pedal.

Figura 4.2 – Fluxograma do *firmware*

Fonte: Produção do autor.

Assim que o pedal é ligado, então todo o circuito será alimentado (MCU, conversores, filtros e pré-amplificador). O microcontrolador, terá um código inicial que segue o fluxograma da figura 4.2. Inicialmente o código deverá ter as inicializações de todas as variáveis e funções utilizadas.

Em seguida, os periféricos I2S, DMA e GPIO são configurados. Após as configurações dos registradores necessários, o MCU irá se comunicar com o conversor AD e receberá o valor digital referente ao sinal de áudio tocado. Esse processo ocorre por meio de interrupções via *software*, ou seja, assim que a amostra for colhetada, é criada uma sinalização indicando que existem amostras a serem processadas. Essas sinalizações acontecem a cada período de amostragem, $T_s = \frac{1}{F_s}$, e são coletadas através dos dois periféricos DMA, que é o controlador responsável também por sinalizar que existem amostras a serem processadas (um controlador DMA serve para receber e outro para transmitir).

Na sequência, com as amostras prontas para o processamento, o código verifica se existe alguma interrupção externa, que vem do usuário através de botões. Caso haja algum botão pressionado, então o programa se deslocará para a rotina de efeitos desejada, e as amostras colhidas serão modificadas de acordo com a rotina escolhida.

Por fim, o periférico DMA transmite as amostras processadas ao conversor DA, e assim termina o papel do código no processamento. O código trabalha em um processamento chamado de "amostra por amostra", ou seja, a cada período de amostragem, todo o processo descrito anteriormente é executado. Existem outras formas de executar esse código, como por

exemplo, o processamento em blocos. Esse tipo de processamento tenta aliviar a CPU de trabalho excessivo, entretanto, insere um atraso entre a saída e entrada, uma vez que, deve-se esperar por um determinado número de amostras serem colhidas.

Para este projeto, o processamento por amostra mostrou-se excelente, pois as demasiadas interrupções não alteram no processamento em tempo real.

4.2 PROTOCOLO DE COMUNICAÇÃO

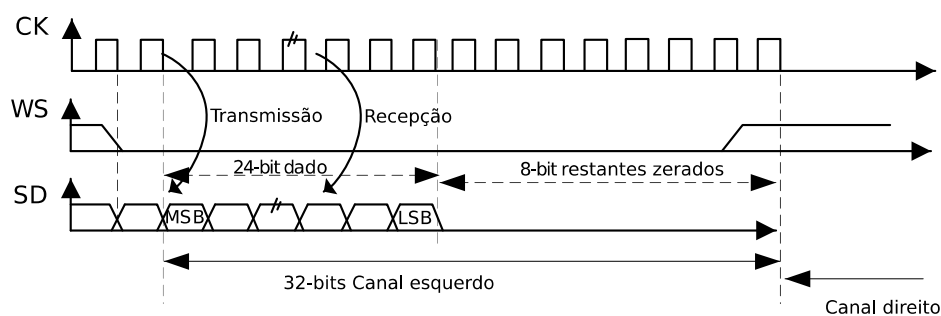
Esta seção dedica-se ao processo de comunicação entre os conversores e a CPU.

Essa comunicação é obtida através do protocolo I2S, *I²S* ou *Inter-IC Sound*. Este protocolo foi criado pela *Philips Semiconductor* com o propósito de comunicar linhas de dados de áudio digital com o formato de PCM (*Pulse Code Modulation*).

Nesse microcontrolador em especial, esse protocolo deriva do perifério SPI com o destino a transferir dados de áudio entre um microcontrolador/DSP para os *codecs*, podendo operar em tempo real ou transferindo dados armazenados na memória do microprocessador (STMICROELECTRONICS, 2014).

A transmissão I2S, utiliza três linhas de *clock* para a transmissão síncrona e serial de dados, os mesmos podem ser vistos na figura 4.3.

Figura 4.3 – Protocolo de comunicação *i²S*



Fonte: Adaptado de (STMICROELECTRONICS, 2014).

A seguir estão os três *clocks* detalhados.

- SCK: essa linha de *clock* é a principal e responsável por sincronizar as outras linhas de *clock*.
- SD: os dados são transmitidos na linha SD (*Serial Data*), no formato *little endian* (*MSB first*) e em complemento de dois. O tamanho dos dados estão geralmente entre 16 - 32 bits.

Os dados são sincronizados pela borda de subida e/ou descida do CK para o transmissor, e pela borda de descida do CK para o receptor. Os dados representam som digital *stereo*, então cada amostra contém duas palavras (*words*). A amostra do canal direito e a amostra do canal esquerdo.

- WS: *Word select* é a linha de controle. Essa linha determina se a palavra sendo enviada é do canal da direita ou da esquerda. Esse sinal também determina o começo e o final dos dados. WS é sincronizado através da borda de descida e/ou subida do CK e antecede o MSB por um período do CK

A configuração do protocolo de comunicação foi escolhida como *Full Duplex*, ou seja, enquanto o controlador DMA recebe dados do ADC, pode, ao mesmo tempo, enviar dados ao DAC. Dessa forma, pode-se transmitir áudio em tempo real. Além disso, o microcontrolador será o mestre nesse protocolo de comunicação, enquanto os conversores serão escravos, isso significa que o MCU proverá todos os *clocks* necessários.

4.3 CONTROLADOR DMA

Devido as evoluções tecnológicas, o uso de processamento digital de sinais, especialmente de áudio, tem sido aplicado em projetos com alta qualidade de áudio. Os métodos de processamento digital de sinais, permitem substituir os efeitos analógicos por digitais, e, além disso, permite criar efeitos que são de difícil aplicação no âmbito analógico. Uma importante vantagem é a oportunidade de ter vários efeitos de áudio em um único sistema. Além disso, hoje pode-se obter efeitos de áudios com microcontroladores de baixo custo não sendo, necessariamente, processadores digitais de sinais (DSP) (BUŚ; JęDRZEJEWSKI, 2015).

A escolha de um microcontrolador atuando como DSP torna-se possível para esse projeto devido ao controlador DMA, do inglês, *Direct Memory Access*. Esse controlador é usado para transferir dados com alta velocidade entre periféricos para memória e de memória para memória sem a interferência da CPU (STMICROELECTRONICS, 2015).

Também, estudos mostram que esse microcontrolador desempenha muito bem a função de DSP tanto para o ensino de algoritmos de processamento digital de sinais, quanto para suas aplicações (WICKERT, 2015) e (WEBSTER; LENOST; KLANG, 2013).

A tabela a seguir resume alguns parâmetros do periférico DMA. O controlador possui dois canais, permitindo a conversão *full-duplex* (enquanto um controlador recebe algumas amostras, o segundo envia outras amostras). Também, os DMAs podem ser utilizados com 8, 16 e 32bits.

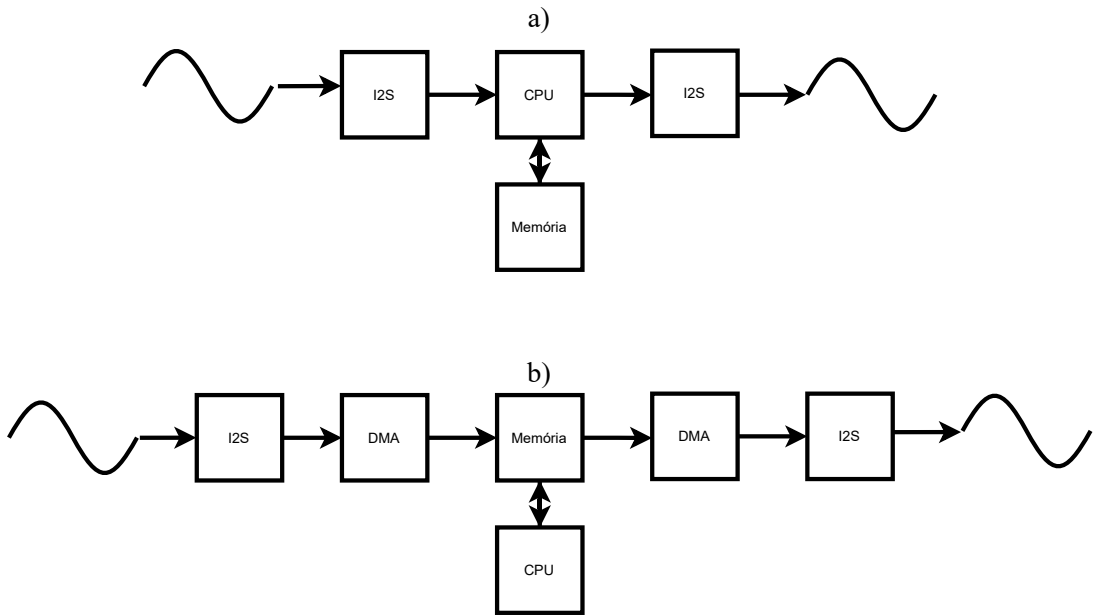
Tabela 4.1 – Parâmetros DMA

Número de controladores	2
Canais	16 (8 em cada controlador)
Tem FIFO?	Sim
Mode	Normal ou Circular
Tamanho de bits para transferência	Byte, Half word e Word
Interrupções	Via Software
Interrupções utilizada	DMA Half Transfer, DMA Transfer complete

Fonte: Adaptado de (STMICROELECTRONICS, 2015).

A figura a seguir demonstra o processo de transmissão de dados do tipo periférico (ADC) para memória (RAM), utilizando DMA na arquitetura desse MCU.

Figura 4.4 – Arquitetura DMA - STM32F407



Fonte: (STMICROELECTRONICS, 2015)

A figura 4.6a representa um sistema digital de aquisição de dados sem DMA. Geramente, utiliza-se métodos como *polling* e interrupção para processar as amostras. Nos casos onde se usa *polling*, a CPU espera até que a conversão do sinal tenha acontecido e não pode realizar outras tarefas, tornando o método muito ineficiente para aplicações que demandam alto desempenho.

Nos casos onde se utiliza interrupções, a CPU tem liberdade para executar outras tarefas e cálculos, e só processa a amostra incidente quando a rotina de interrupção for acionado. Esse método é mais eficiente, porém, quando se trabalha com um número demasiado de interrupções, a CPU acaba ficando sobrecarregada, pois a mesma também é responsável por transmitir os dados para a memória.

A figura 4.6b apresenta o modo mais eficiente de transmitir amostras em um curto espaço de tempo. Este método utiliza o controlador DMA. Esse controlador, é responsável por transmitir várias amostras e blocos de amostras entre periféricos e de memória para memória, sem a interferência da CPU. Ou seja, o processo também ocorre por interrupções, porém nesse caso, a CPU não tem que lidar com as transfêrencias da amostra entre os periféricos, ela apenas necessita realizar os cálculos necessários.

Neste microcontrolador, o controlador DMA seta duas sinalizações (*flags*), em um único período de amostragem, para manipular as duas interrupções: *half transfer* e *cplt transfer*.

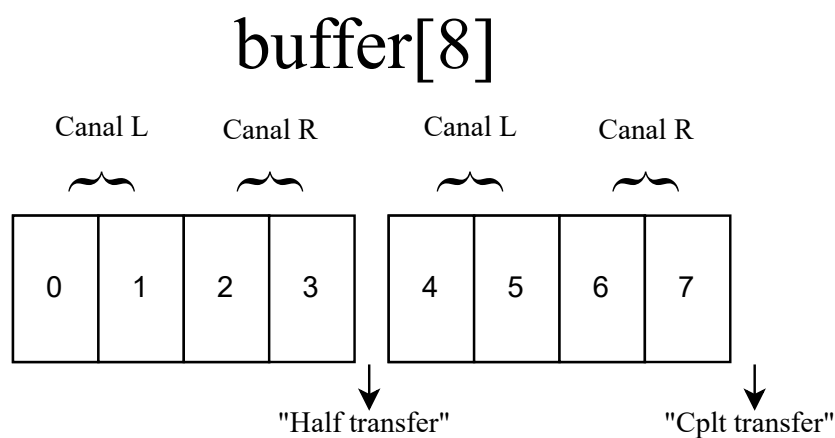
A primeira *flag* indica que metade do **buffer** foi transmitido, seja de um periférico para memória, ou de memória para periférico. A segunda indica que o processo terminou. Usualmente, a utilização dessas *flags* são importantes para que a CPU não acesse a mesmo endereço de memória que o DMA. Embora o DMA seja utilizado naturalmente em processamentos por blocos, pode-se utilizar um vetor de duas posições por exemplo, contendo o canal direito e o esquerdo, caracterizando um processamento por amostra.

4.4 DOUBLE BUFFER

A técnica *double buffer* ou *ping-pong buffers* é muito utilizada em processamento de imagens e de áudio. Essa técnica permite, ao mesmo tempo que esteja sendo processado um *buffer*, outro *buffer* esteja sendo recebido ou enviado.

A figura a seguir demonstra como as notas musicais são convertidas e transmitidas nesse MCU.

Figura 4.5 – *Double buffer*

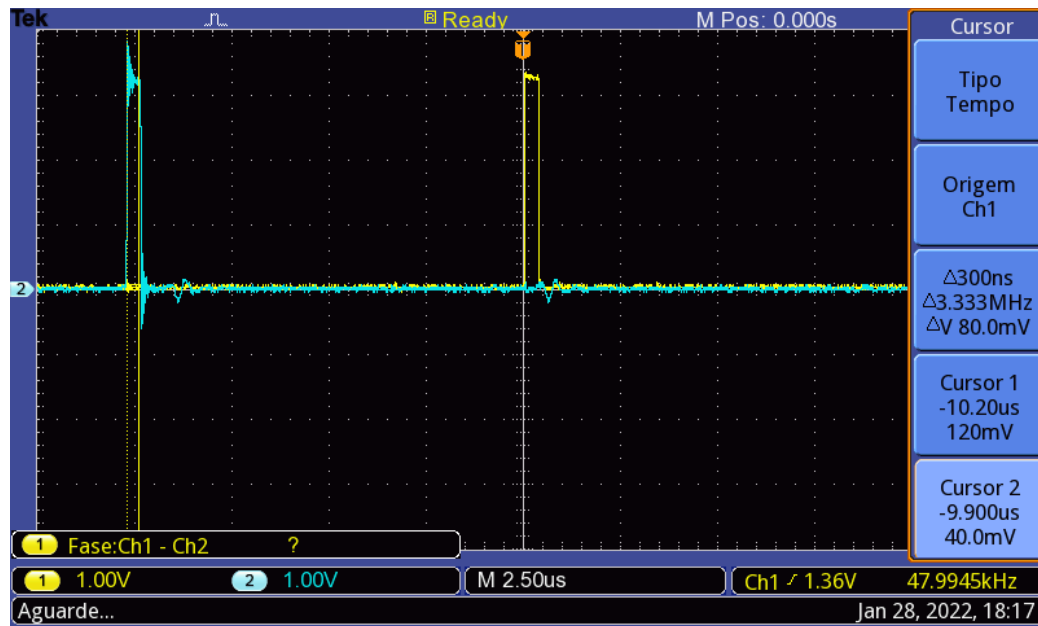


Fonte: Produção do autor.

O processo ocorre utilizando um bloco de oito amostras. Cada amostra (0,1,2..6,7) é definida como sendo de 16bits, pois o periférico I2S só transmite com esse tamanho. Quando o canal um do DMA recebe as quatro primeiras amostras armazenadas na memória, oriundas do periférico I2S, o mesmo gera a *flag "half transfer"* e transmite os canais "L" e "R", que são abreviações para o canal esquerdo e direito. Como o ADC transmite uma amostra com 24bits, então há a necessidade de utilizar duas amostras de 16bits para a manipulação. Com essa *flag* setada, a CPU pode executar a rotina de efeitos programadas. O tempo que essa *flag* fica em nível lógico alto será medido e mostrado posteriormente. Deve-se salientar que durante esse período, as amostras 4,5,6 e 7 estão sendo transmitidas pelo segundo canal do controlador DMA para o periférico I2S, que efetua a transmissão para o DA.

Após a rotina de interrupção "*half transfer*", o segundo canal do DMA agora recebe outras quatro amostras oriundas do conversor AD. Após a transmissão dessas amostras para a memória, o canal dois do DMA gera a *flag "cplt transfer"* e a CPU acessa outra rotina de interrupção. Em resumo, metade do *buffer* da figura anterior é processado durante uma interrupção, e a outra metade é enviada ao conversor DA. Após um período de amostragem, essa ordem é invertida. Esse processo é cíclico e ocorre indefinidamente.

A figura a seguir demonstra esse conceito de forma clara. A onda em azul representa quando a sinalização "*half transfer*" foi setada. Dessa forma, a CPU executa o algoritmo, que nesse caso, apenas copia os dados de entrada para a saída. A onda em amarelo representa a sinalização "*cplt transfer*" e executa a mesma rotina. Percebe-se que cada onda separadamente apresenta um período de 48kHz, e no total tem-se 96kHz de amostragem (somando as duas). As rotinas da figura a seguir possuem um tempo total de 300ns cada.

Figura 4.6 – *Double buffer* na prática

Fonte: Produção do autor.

4.5 SÍNTESE DO CAPÍTULO

Neste capítulo foi apresentado todo o projeto do ponto de vista do código. Alguns conceitos devem ser bem entendidos: Neste projeto o periférico I2S atua como *full duplex* recebendo e transmitindo dados ao mesmo tempo; as amostras são transmitidas em um vetor de *16bits* contendo oito amostras; as amostras são armazenadas na memória através do controlador DMA que executa o papel da CPU. Com o entendimento desses conceitos, pode-se estudar os algoritmos para os efeitos.

5 DISTORÇÃO

Neste capítulo será feito um estudo e implementação de um algoritmo para a realização de efeitos como distorção e *overdrive*.

5.1 PROCESSAMENTO NÃO LINEAR

O efeito de distorção pode ser descrito como a modificação do som usando um processamento não linear do sinal. Esse efeito produz não linearidades no som original, através de injeção de componentes harmônicas no sinal de entrada (ZOLZER, 2002). Esse efeito surgiu através dos primeiros pedais de distorção que eram construídos utilizando válvulas. Nos anos 50, descobriu-se que o fato de aplicar uma tensão sobre a válvula, de modo que a tensão de operação da mesma fosse excedida, saturando a saída, produzia um som característico com bastante aplicações no meio musical (PAKARINEN; Yeh, 2009).

Dentro dos efeitos de guitarra não lineares, tem-se, de forma genérica, dois efeitos: distorção e *overdrive*.

- **Overdrive:** De forma geral, o efeito de *overdrive* é caracterizado por sinais com baixa amplitude de entrada continuarem lineares, porém com o aumento da amplitude de entrada, o sinal passa para a região não linear. Quando a amplitude do sinal de entrada chega ao seu máximo, então a saída do sistema satura para um valor definido (ZOLZER, 2002).
- **Distorção:** Os efeitos de distorção similares aos de guitarristas como Jimi Hendrix é baseado nas simulações das válvulas dos amplificadores e circuitos analógicos. Os guitarristas contemporâneos de Hendrix utilizam circuitos analógicos, geralmente, combinando transistores BJT em emissor comum, o que provocava pouca distorção do sinal, para baixas amplitudes de entrada (com uma adição de harmônicas de segunda ordem). Sinais com alta amplitude eram distorcidos e recebiam conteúdo harmônico tanto de ordem par quanto de ordem ímpar. Essa não linearidade pode ser simulada digitalmente através dos efeitos chamados de *asymmetrical clipping* (ZOLZER, 2002).

De forma geral, os efeitos de *overdrive* são simétricos, ou seja, tanto o sinal positivo quanto o negativo são distorcidos da mesma forma. Os efeitos de distorção também podem ser simétricos, mas também há a categoria dos efeitos assimétricos, onde a parte positiva do sinal é distorcida de uma maneira diferente do que a negativa.

5.2 MODELOS

Sabendo-se então que o sinal de distorção consiste basicamente na injeção de harmônicos no sinal, pode-se projetar circuitos (analógicos ou digitais) com a finalidade de distorcer o sinal, e ainda sim conter uma boa qualidade de áudio.

Segundo Pakarinen e Yeh (2009), tais circuitos são constituídos por válvulas e dispositivos de estado sólido. As válvulas acabam produzindo efeitos com uma maior qualidade em comparação com circuitos transistorizados, o que evidencia o fato de que grande parte do comércio de válvulas existente nos dias de hoje, é destinado à dispositivos de áudio (BARBOUR, 1998).

Contudo, o mercado de válvulas é reduzido, seus dispositivos (em sua grande maioria amplificadores) são caros e de difícil acesso, além dos problemas comuns inerentes as válvulas, como: tamanho elevado, baixa durabilidade, preço alto e alto consumo.

Dessa forma, outras formas de se obter distorção em sinais de áudio surgem. Dentre essas técnicas citam-se algumas a seguir.

5.2.1 *Waveshaping*

A técnica de *waveshaping* consiste em aplicar um modelo matemático não variável, relacionando os dados de saída com a entrada. Essa técnica é muito utilizada em microcontroladores e DSPs pois, dependendo do modelo, não exige computação elevada da CPU. Um dos modelos mais conhecidos foi apresentado por Araya e Suyama (1996) em sua patente.

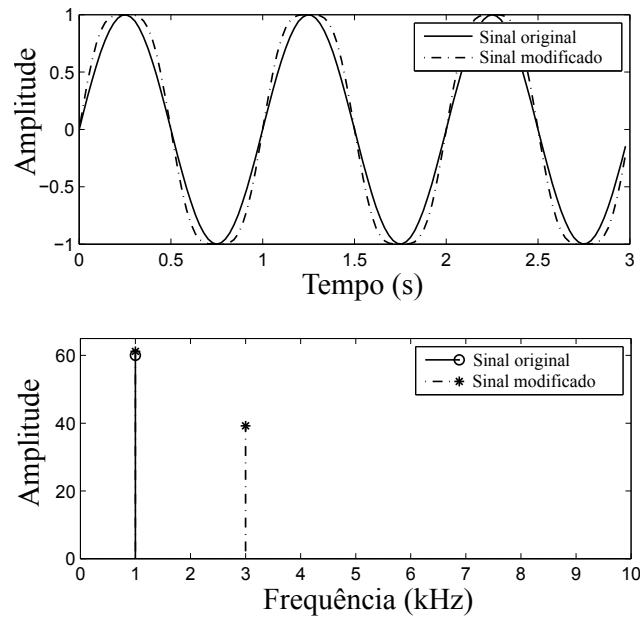
A equação a seguir apresenta o modelo apresentado na patente.

$$y = \frac{3x}{2} \left(1 - \frac{x^2}{3}\right) \quad (5.1)$$

Na figura 5.1, tem-se a representação no domínio do tempo e frequência para a equação (5.1). No domínio do tempo, pode-se perceber a diferença entre o sinal original e o sinal através da equação. O sinal original é uma senóide de $1kHz$ com amplitude unitária.

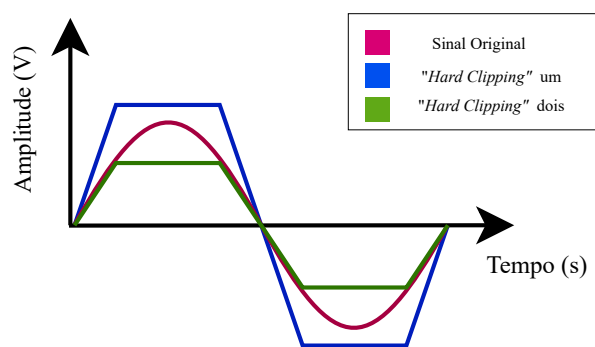
Em seguida, pode-se observar a inserção de harmônicos no sinal, especificamente, a terceira harmônica inserida. Segundo Zolzer (2002), a distorção proposta por Araya e Suyama (1996) é suave, e para se ter mais distorção, pode-se aplicar várias vezes a mesma equação.

Figura 5.1 – Distorção por Araya e Suyama (1996)



Fonte: Adaptado de (OLIVEIRA; BARRETO; PASQUAL, 2013).

Outra técnica de *waveshaping* conhecida é chamada de *hard clipping*. A técnica consiste em cortar o sinal em uma determinada amplitude, podendo ser maior ou menor do que a amplitude máxima do sinal original. Essa técnica pode ser observada a seguir.

Figura 5.2 – Distorção *hard clipping*

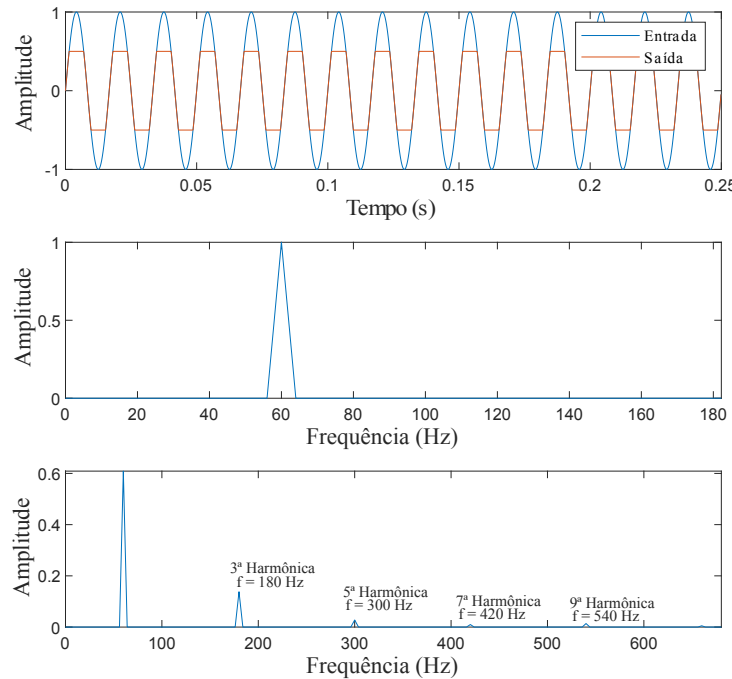
Fonte: Produção do autor.

Uma maneira de implementar esse efeito, de forma analógica, é projetar um circuito onde seja adicionado um ganho elevado ao sinal de entrada, até que o sinal se aproxime da tensão de alimentação e saturar, ou então, utiliza-se diodos na realimentação de amplificadores operacionais para ceifar o sinal original. A implementação no domínio digital procura seguir uma equação de diferenças e pode ser encontrada a seguir.

$$y[n] = \begin{cases} A, & |x[n]| \geq A \\ x[n], & |x[n]| < A \end{cases} \quad (5.2)$$

Onde "A" é um parâmetro controlado pelo usuário, e é lido através do conversor AD interno do microcontrolador. A figura a seguir apresenta uma senóide de entrada com frequência de 60Hz e amplitude unitária.

Figura 5.3 – Distorção com corte em 50% da entrada



Fonte: Produção do autor.

A saída está apresentada na primeira imagem, como esperado, existe o corte da mesma. A amplitude de corte foi selecionada em 50%. Logo abaixo, pode-se observar a transformada de *fourier* do sinal de entrada, com apenas um harmônica em 60Hz. Em seguida, observa-se o conteúdo harmônico do sinal distorcido. Se o parâmetro de corte for aumentado, então a saída tenderá a se aproximar da entrada, as harmônicas de ordem ímpar diminuirão e a harmônica fundamental aumentará. Com a diminuição da amplitude do corte, a harmônica fundamental diminuirá e as outras aumentarão.

5.2.2 Modelo digital

Este método consiste em criar modelos digitais através de resultados de modelos físicos. Diferentemente do modelo anterior, um modelo digital leva em conta as não linearidades

inerentes aos circuitos analógicos. Trabalhos como (ZEKI, 2015) e Overton (2006) buscam equacionar funções que caracterizem a distorção inserida, especificamente em amplificadores válvulados, onde o efeito foi observado inicialmente.

Overton (2006) emula digitalmente um circuito transistorizado de distorção conhecido como *Fuzz Face*. O autor usa inicialmente simulação *SPICE* para entender como os equações não lineares, relacionando entrada e saída, são calculadas. Em seguida, há uma análise dos efeitos da troca de determinados componentes no circuito. Utilizando o modelo *Ebers-Moll* para os transistores, um modelo digital é criado.

De acordo com o autor, o modelo digital é uma excelente aproximação, embora não carregue todas as não linearidades presentes. Entretanto, os cálculos são elaborados considerando um processamento *off-line* e as equações levam um grande período para serem efetuadas.

5.3 SÍNTESE DO CAPÍTULO

Este capítulo apresentou possíveis técnicas para emular efeito de distorção em processadores de sinais. Para este trabalho, buscou-se implementar a equação de diferenças (5.2), pois adiciona um alto conteúdo harmônico e é facilmente implementada. Os resultados serão apresentados na seção 9.

6 DELAY

Esta seção é dedicada ao estudo e implementação dos efeitos de guitarra baseados em atraso, especificamente atrasos por números inteiros. Procura-se entender o funcionamento dos efeitos no mundo analógico (análise qualitativa) e busca-se uma forma da implementação dos mesmos no âmbito digital (análise quantitativa).

6.1 ANÁLISE QUALITATIVA

Fisicamente, o efeito de atraso pode ser presenciado em todos os ambientes, seja por meio de uma rápida repetição da onda sonora, ou até mesmo um eco. Essas características surgem da sobreposição do som, que trafega diretamente da fonte sonora para o ouvinte, em conjunto com as ondas sonoras refletidas e atenuadas ao longo do percurso. A diferença de tempo entre o sinal direto e as suas reflexões define o que são conhecidos na física acústica como eco e reverberação. Se o sinal atrasado no tempo levar um intervalo de tempo superior a $100ms$ para chegar ao ouvinte, tem-se o eco, e se levar um intervalo de tempo menor, tem-se a reverberação.

Acusticamente, tais efeitos são comuns em salas de concerto e teatros, e na maioria das vezes causa um som desagradável para a percepção da plateia, necessitando a eliminação dos mesmos. Entretanto, no meio musical, o atraso do som é inserido dentro de solos de guitarra de forma proposital.

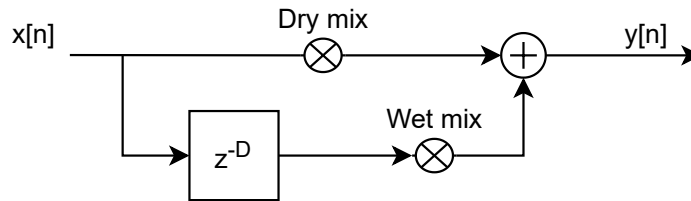
Os circuitos analógicos que criam essa característica sonora utilizam uma técnica chamada de BBD - *Bucket Brigade Device*, que foi desenvolvida em 1960 (RAFFEL, 2010). O nome dessa topologia vem da analogia de um grupo de pessoas passando um balde (*bucket*) de água através de uma linha, para apagar uma chama. Similarmente, o que este dispositivo procura realizar é passar o sinal de áudio, amostrado utilizando transistores MOS, de um estágio para outro, até chegar ao final da linha. Entretanto, cada estágio é responsável por adicionar um atraso.

Embora os circuitos integrados que utilizam essa técnica não utilizam uma quantização do sinal, ainda sim é efetuado uma amostragem do mesmo. Sendo assim, em conjuntos com esses CIs, encontra-se filtros *antialiasing* e de reconstrução de segunda até quarta ordem (HUOVILAINEN, 2005). Entretanto, tal topologia é considerada como pedais de atraso analógicos, pois utilizam componentes analógicos. Para simular um sistema de atraso totalmente digital, pode-se fazer uma analogia dos estágios do BBD em conjunto com a teoria do processamento digital de sinais.

6.2 ANÁLISE QUANTITATIVA

Pode-se implementar digitalmente os atrasos utilizando os conceitos de filtro FIR e IIR. O diagrama de blocos da figura 6.1 demonstra a representação do efeito de atraso como um filtro FIR.

Figura 6.1 – Diagrama de blocos para *delay* FIR



Fonte: Adaptado de (ZOLZER, 2002)

Percebe-se que a saída do sistema recebe tanto o sinal de entrada, quanto o mesmo sinal atrasado no tempo e atenuado. O filtro FIR representa apenas um único *delay*.

A equação de diferença da figura 6.1 é mostrada a seguir.

$$y[n] = Dry * x[n] + Wet * x[n - D] \quad (6.1)$$

Onde n é a amostra do sinal, " y " a saída, " x " é o sinal de entrada amostrado, " wet " é o coeficiente de reflexão e também representa o sinal atrasado, " dry " é o coeficiente que atenua o sinal direto da fonte sonora ao ouvinte e D é valor de atraso escolhido.

Pode-se usar a transformada z da equação anterior para entender o funcionamento do atraso no domínio da frequência.

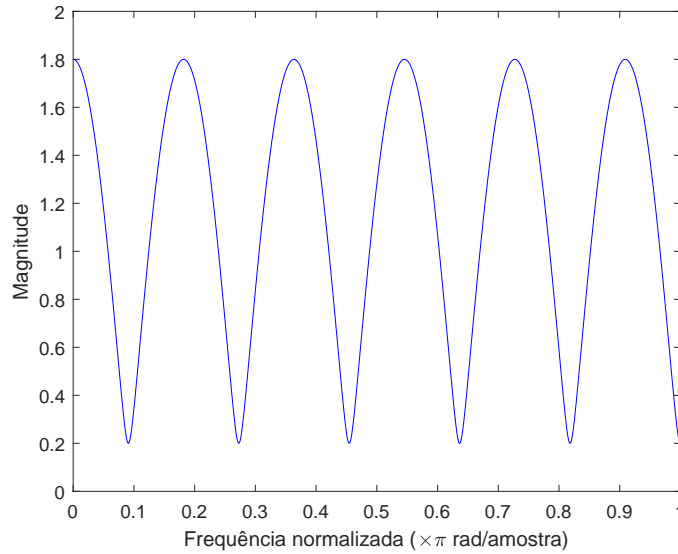
$$H(z) = Dry + Wet * z^{-D} \quad (6.2)$$

Substituindo $z = e^{j\omega}$, e utilizando a fórmula de *Euler*, pode-se obter o módulo da resposta $H(z)$, como é observado a seguir.

$$|H(e^{j\omega})| = \sqrt{1 + 2 * Wet * \cos(\omega D) + Wet^2} \quad (6.3)$$

Para exemplificar, pode-se escolher o parâmetro *Wet* como 0,8. A figura a seguir demonstra a resposta em frequência do filtro. A partir do formato da resposta é que se origina o nome de filtro (pente).

Figura 6.2 – Resposta em frequência para o efeito *Delay*



Fonte: Produção do autor.

Os picos da figura anterior acontecem quando $\omega_k = \frac{2k\pi}{D}$, com $k = 0, 1, 2$ e *etc* (ORFANIDIS, 2010). Dessa forma, a amplitude tem um máximo dado na equação a seguir.

$$|H(e^{j\omega})| = \sqrt{(1 + 2 * Wet + Wet^2)} = 1 + Wet \quad (6.4)$$

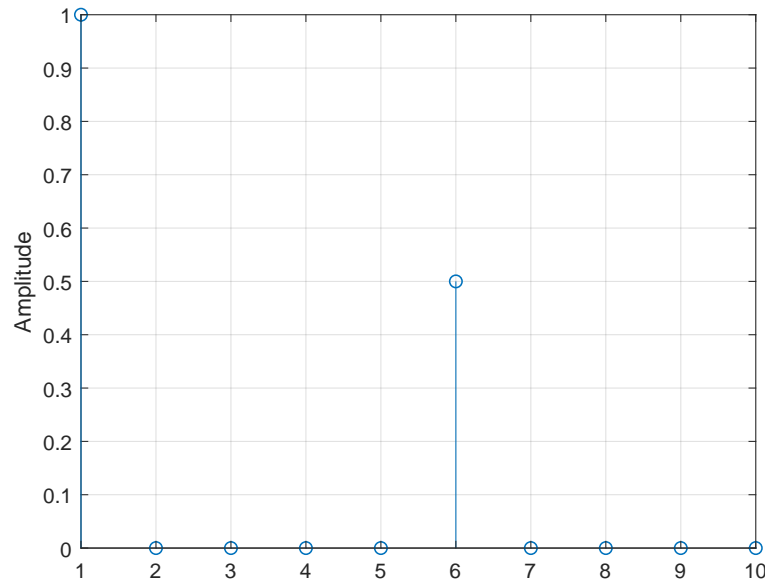
Na figura, observa-se que os picos são $1+0,8 = 2,8$.

Por outro lado, o módulo da resposta em frequência será mínima quando $\omega_k = \frac{(2k+1)\pi}{D}$, com $k = 0, 1, 2$ e *etc* (ORFANIDIS, 2010). Dessa forma, a amplitude terá um mínimo dado na equação a seguir.

$$|H(e^{j\omega})| = \sqrt{(1 - 2 * Wet + Wet^2)} = 1 - Wet \quad (6.5)$$

A resposta ao impulso do filtro pente é visualizada a seguir. Com um tamanho de dez amostras, um atraso (D) de 5 amostras, parâmetro *dry* igual a um e parâmetro *wet* igual a 0,5, pode-se observar o comportamento do filtro, e como os parâmetros influenciam na cópia atrasada. O algoritmo responsável por criar essa resposta ao impulso deve ser capaz de modificar esses parâmetros de acordo com a modificação efetuada pelo usuário.

Figura 6.3 – Resposta ao impulso de um filtro pente



Fonte: Produção do autor.

De acordo com as figuras 6.1 e 6.3, para a implementação em tempo de real de um algoritmo que simula o efeito de atraso, a variável de saída receberá sempre a amostra atual convertida pelo conversor PCM1802, em conjunto com uma amostra passada. A amostra passada é facilmente manipulada utilizando o conceito de **linha de atraso**. Tal conceito consiste basicamente em um vetor que irá armazenar as amostras de áudio. Tal vetor, receberá uma nova variável a cada período de amostragem. Como uma forma de implementação, utiliza-se uma técnica chamada de *buffer* circular. Além disso, o tamanho do *buffer* é um parâmetro importante pois utiliza uma grande quantidade de memória do microcontrolador, variando de acordo com a quantidade de atraso escolhido pelo usuário. A seguir, encontra-se uma implementação dessas características no MCU escolhido.

6.2.1 Armazenamento

Para um processamento *offline*, pode-se ter acesso a qualquer amostra em qualquer tempo. Entretanto, para o processamento em tempo real, deve-se utilizar vetores (também chamado de *buffers*) para armazenar as amostras passadas, que são inseridas a cada intervalo de amostragem ($\frac{1}{F_s}$). O tamanho desse vetor é calculado seguindo a equação (6.6), com um atraso de 100ms e frequência de amostragem de 96kHz.

$$BUFFER_LENGTH = D * F_s = 9600 \quad (6.6)$$

A equação a seguir demonstra que para um atraso de 100ms, deve-se ter na memória RAM um vetor de tamanho 9600 que ocupa um espaço de $9600 \times 4 = 38,4\text{KB}$ (de forma geral, variáveis do tipo `int` em linguagem C possuem um tamanho de 2 a 4 kB). Tal valor deve ser um dos parâmetros para a escolha do microprocessador, pois o mesmo indica a possibilidade de armazenamento das amostras. Para os efeitos propostos, o MCU provê armazenamento para, até, 300ms de atraso, tempo suficiente para o efeito. A tabela a seguir demonstra o uso de memória total do projeto (levando em conta todos os quatro efeitos com um atraso de 100ms para o efeito de *delay* e um atraso de 15ms para o efeito *flanger*).

Tabela 6.1 – Configuração do DMA

Memória	Tamanho	Espaço usado
RAM	128KB	42,22KB
FLASH	1024KB	16,15KB

Fonte: Produção do autor.

6.2.2 Buffer circular

Antes de entender o conceito por trás do *buffer* circular, deve-se entender como é calculado um filtro FIR e a problemática por trás dos mesmos quando são utilizados em tempo real.

A equação de um filtro FIR é dada a seguir.

$$y[n] = \sum_{k=0}^N b_k x[n-k] \quad (6.7)$$

Para um filtro de ordem três (por exemplo), a expressão anterior pode ser reescrita como a seguir.

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + b_3 x[n-3] \quad (6.8)$$

Os termos $x[n-k]$ armazenam as amostras passadas. No primeiro momento, quando "n" é igual a zero, então $y[0] = b_0 x[0]$ apenas. Em seguida, o conteúdo da amostra $x[0]$ deverá ser atualizado para um novo espaço ($x[n-1]$) para que o espaço do vetor $x[0]$ esteja livre para receber a amostra proveniente do ADC. E assim por diante.

O algoritmo a seguir demonstra uma simples implementação para processar os dados em uma rotina FIR (WELCH; WRIGHT; MORROW, 2006).

Algoritmo 1: Rotina para filtro FIR

Entrada: Valor atual do ADC

Saída: Valor filtrado para o DAC

início

float b[N+1];

float x[N+1]

float y = 0;

para ($i=0; i \leq N; i++$) **faça**

| $y = y + x[i] * b[i];$

fim

para ($i=N; i > 0; i--$) **faça**

| $x[i] = x[i-1];$

fim

fim

retorna y

O segundo laço "for" realiza a seguinte operação.

$$\begin{aligned} x[2] &\rightarrow x[3] \\ x[1] &\rightarrow x[2] \\ x[0] &\rightarrow x[1] \end{aligned} \tag{6.9}$$

Ou seja, o algoritmo perde muito tempo apenas deslocando as amostras ao longo do vetor, para que sempre o primeiro índice do vetor receba o valor do conversor AD. Segundo Orfanidis (2010), isso é uma técnica muito ineficiente, pois para efeitos de áudio, os vetores são extensos. Para contornar esse problema, utiliza-se o conceito de *buffer* circular.

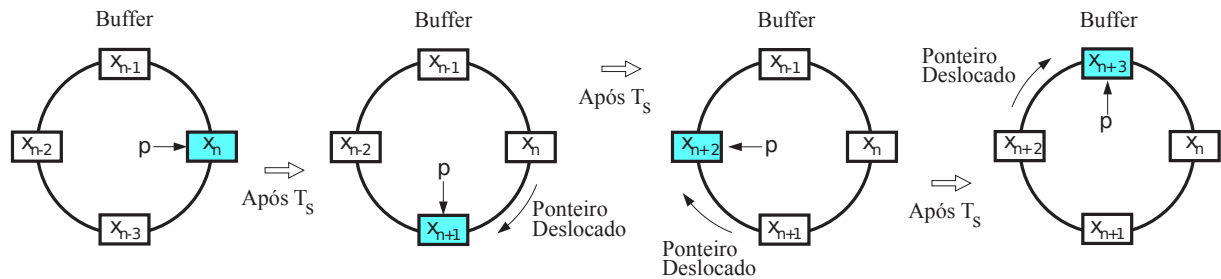
Para este processo, utiliza-se ponteiros para gravar as posições ao longo da linha de atraso. Tal ideia está representado na figura 6.4.

Na figura 6.4, no momento inicial, o ponteiro "p" aponta para o primeiro índice do *buffer* (figura na extrema esquerda). Para acessar o conteúdo " x_n ", pode-se utilizar $buffer[p]$ e processar os algoritmos desejados. Após terminar a rotina, o ponteiro é incrementado em uma unidade, e após o próximo período de amostragem, pode-se acessar o segundo índice do *buffer*, utilizando novamente $buffer[p]$. Esse processo é evidenciado na figura, e é cíclico, originando o termo "circular" no nome.

Algumas características podem ser observadas: não existe mais o deslocamento de dados dentro do vetor, isso proporciona maior rapidez no processamento dos algoritmos; apenas o

ponteiro irá se deslocar e o índice mais recente está onde o ponteiro está apontando (PIRKLE, 2013).

Figura 6.4 – Buffer circular



Fonte: Adaptado de (ORFANIDIS, 2010)

Para acessar uma amostra passada, por exemplo duas amostras ($D=2$), basta subtrair a posição do ponteiro p por D ($p-D$). Com isso, pode-se resolver a equação (6.1), e somar o sinal de áudio atual com o seu atrasado, abrindo espaço para criar efeitos de áudio baseados em atraso de uma forma eficiente em tempo real, como a seguir.

Algoritmo 2: Efeito de *delay*

Entrada: x = Valor do ADC

Saída: y = Valor para o DAC atual + atrasado

início

```

    ganhoDry = 0,7;
    ganhoWet = 0,7;
    ponteiro=0;
    buffer_length = 9600;
    atraso=buffer[ponteiro];
    y=x*ganhoDry + atraso * ganhoWet
    se Ponteiro++ > buffer_length então
        | ponteiro=0;

```

fim

fim

6.3 SÍNTESE DO CAPÍTULO

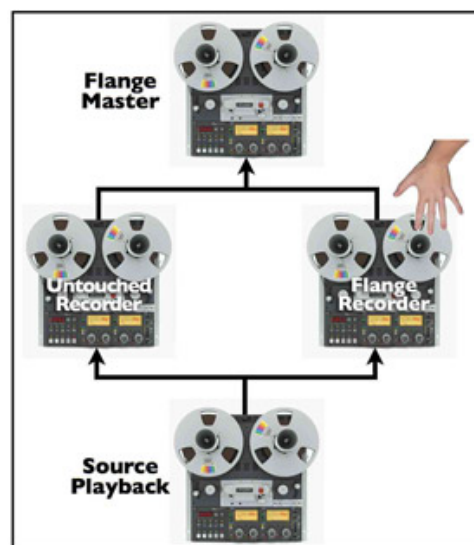
Neste capítulo foi apresentado como efeitos de atrasos por números inteiros são criados no meio digital. Foi apresentado o conceito dos mesmos, formas eficientes de programar, algoritmos e exemplos. A implementação do efeito no sistema embarcado será apresentada na seção 9.

7 FLANGER

Dando sequência aos efeitos sonoros com atraso, este projeto apresenta um caso especial do filtro pente, conhecido popularmente como *flanger*. Este efeito é caracterizado matematicamente por utilizar atrasos com números não inteiros.

A figura a seguir mostra como era obtido o efeito *flanger* abordado neste capítulo.

Figura 7.1 – Efeito *flanger* por gravador de fitas



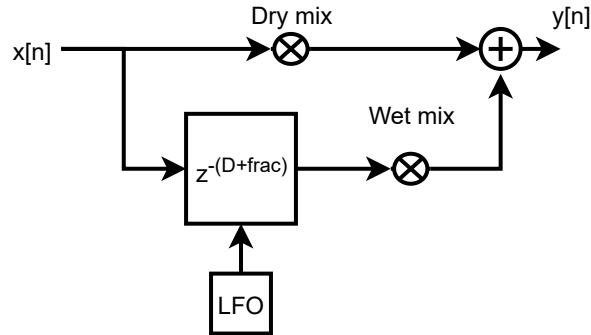
Fonte: (OWSINSKI, 2021)

Segundo (OWSINSKI, 2021), tal efeito surgiu na década de 60. A música nos estúdios era armazenada em dois gravadores, um contendo a música intacta e o segundo gravador armazenava a mesma música com um atraso. Além disso, utilizava-se um terceiro gravador para somar as duas gravações. Ao somar as duas gravações, era percebido o filtro *comb* (notava-se que as duas fontes sonoras estavam em defasagem). Ao pressionar o dedo na parte metálica da gravadora (denominada "flange") o atraso variava lentamente. Surge então o que hoje é conhecido como *flanger*.

Segundo (SMITH, 1984), *flanger* é uma modulação que varia as interferências destrutivas (nulos ou *notches*) de um filtro pente, ao longo do espectro audível. Com o atraso do sinal sendo variado (com um LFO - *Low Frequency Oscillator*), os *notches* (nulos) variam uniformemente, criando assim o som característico chamado *flanger*.

A figura a seguir demonstra o diagrama de blocos para este efeito.

Figura 7.2 – Representação em diagramas de bloco *flanger*



Fonte: Adaptado de (ZOLZER, 2002)

O atraso no tempo, ou em amostras, como é demonstrado na figura, é variável. A equação a seguir apresenta a equação de diferenças com modulação.

$$y[n] = Dry * x[n] + Wet * x[n - (\tau)] \quad (7.1)$$

Onde o parâmetro τ é composto de uma porção inteira e outra fracionária, demonstrada na equação a seguir.

$$\tau = D + \sin(2\pi \frac{f}{F_s} n) \quad (7.2)$$

A seguir há uma descrição dos parâmetros das equações (7.1) e (7.2), associando estes, com os parâmetros nos pedais *comerciais*.

- **D**: Este parâmetro, em produtos comerciais, é a quantidade de atraso total. Para efeitos do tipo *flanger*, o valor para este parâmetro é de no máximo 15ms (ZOLZER, 2002).
- **Dry**: Este parâmetro controla o quanto do sinal sem efeito é aplicado na saída. Em pedais comerciais, o músico não controla este parâmetro.
- Parâmetro "**f**": Frequência do oscilador, o parâmetro f é conhecido como "*speed*" ou "*rate*", controla a variação das interferências destrutivas do filtro pente. Varia entre 0,1Hz à 1Hz.
- **Wet**: Controla o quanto do sinal com efeito é adicionado à saída.

Os parâmetros anteriores, são potenciômetros, que controlam a quantidade de ganho, atraso ou oscilação em circuitos analógicos. Para este projeto, o usuário terá acesso aos potenciômetros para controlar a frequência do oscilador, e do parâmetro *Wet*. A conversão destes valores lidos será realizado pelo conversor AD embarcado ao microcontrolador.

A fim de entender a influência da variação do atraso no sinal, pode-se utilizar a transformada de Fourier da equação (7.1) para o entendimento do problema no domínio da frequência.

As equações a seguir demonstram essa transformação, onde τ representa a modulação senoidal do atraso (equação (7.2)).

$$Y(Z) = X(Z) + X(Z)z^{-\tau} \quad (7.3)$$

A função de transferência é obtida dividindo ambos os lados por $X(Z)$.

$$H(Z) = 1 + z^{-\tau} \quad (7.4)$$

Fazendo a substituição $z = e^{j\omega}$, tem-se a equação a seguir.

$$H(e^{j\omega}) = 1 + e^{-j\omega\tau} \quad (7.5)$$

Pode-se utilizar a fórmula de *Euler* e uma manipulação algébrica para simplificar a expressão (SMITH, 1984).

$$H(e^{j\omega}) = e^{-j\omega\frac{\tau}{2}}(e^{j\omega\frac{\tau}{2}} + e^{-j\omega\frac{\tau}{2}}) \quad (7.6)$$

Reajustando os termos da equação, pode-se obter a equação a seguir, representando a função de transferência desse filtro pente.

$$H(e^{j\omega}) = 2e^{-j\omega\frac{\tau}{2}}\cos(\omega\frac{\tau}{2}) \quad (7.7)$$

O módulo pode-se obter como a equação a seguir.

$$|H(e^{j\omega})| = 2\cos(\omega\frac{\tau}{2}) \quad (7.8)$$

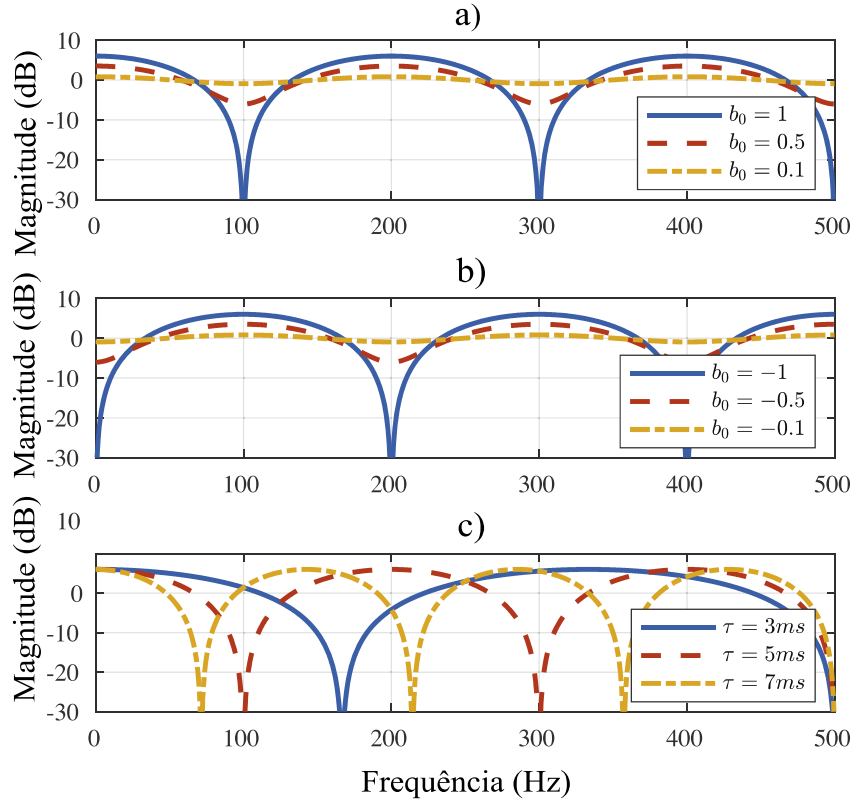
A figura a seguir apresenta a resposta em frequência do filtro pente com atrasos variáveis, para diferentes valores de atrasos (o parâmetro "*dry*" foi substituído por b_o).

A figura 7.3a apresenta a variação do parâmetro *dry*. A variação do mesmo influencia no quanto acentuado os nulos são. Com um valor próximo de zero, então não há mais a existência do filtro *comb* pois a saída receberá apenas uma amostra passada. Com o aumento do parâmetro, então os nulos começam a se acentuar. A figura 7.3b apresenta o mesmo conceito, porém com valores negativos. A diferença consiste apenas na posição em que ocorrem os nulos.

A figura 7.3c apresenta o caso especial do *flanger*. Percebe-se que o aumento do atraso introduz um número maior de nulos em determinadas frequências. Como o atraso varia de forma senoidal, então os nulos acompanham essa modulação. O resultado dessa modulação

pode ser ouvido como o som de um avião passando sobre um observador, ou uma ambulância com a sirene ligada se aproximando e, posteriormente, se afastando de um observador.

Figura 7.3 – Resposta em frequência filtro *comb* com atraso variável



Fonte: Adaptado de (WRIGHT; VÄLIMÄKI, 2021).

Para a implementação deste efeito, será utilizado os mesmos conceitos da seção 6, com algumas modificações. Como este efeito trabalha com atrasos menores, então o tamanho do vetor necessário é reduzido. Considerando o atraso máximo de $15ms$, então o tamanho do vetor utilizado é calculado a seguir.

$$\text{Tamanho do vetor} = 15ms \times 96kHz = 1440 \quad (7.9)$$

O espaço necessário para armazenamento é $4 \times 1440 = 5750kB$.

Para o *buffer circular*, deve-se ter dois ponteiros, o primeiro ponteiro armazenará a posição no vetor da amostra atual, e é comumente chamado de ponteiro de escrita (*write*). O segundo ponteiro, chamado de ponteiro de leitura (*read*), aponta para as amostras passadas.

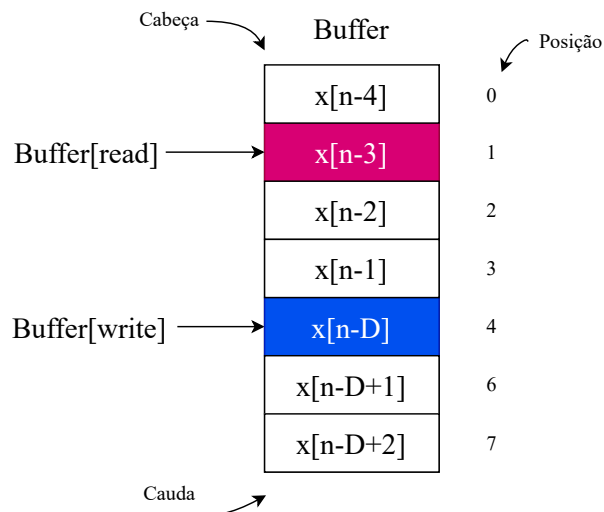
O uso de dois ponteiros se faz necessário pois o atraso é variável, dessa forma, deve-se ter um fixo como referência e outro variando ao longo do vetor.

A figura a seguir demonstra o conceito da implementação. O espaço da memória em azul representa a amostra de dado mais antiga (que será substituída pelo valor do ADC atual). Esse espaço é sobrescrito a cada período de amostragem. Os espaços anteriores a $x[n - D]$ são as

últimas amostras acessadas. O espaço em rosa representa a amostra necessária para cumprir um determinado atraso.

Como exemplo na figura, utiliza-se um vetor de tamanho sete. Em dado momento, o ponteiro "*write*" aponta para a posição quatro, e o ponteiro "*read*" aponta para a posição um. A diferença entre posições é o atraso ouvido ($D=4-1=3$). Por fim, a cada período de amostragem as amostras são substituídas por novos valores, e os ponteiros são incrementados. Assim que os ponteiros atingirem a "cauda" do vetor, eles voltam para a "cabeça" enfatizando o conceito de *buffer* circular. No caso do efeito *delay*, o espaço da memória em rosa é mantido fixo e no caso do efeito *flanger* o mesmo varia.

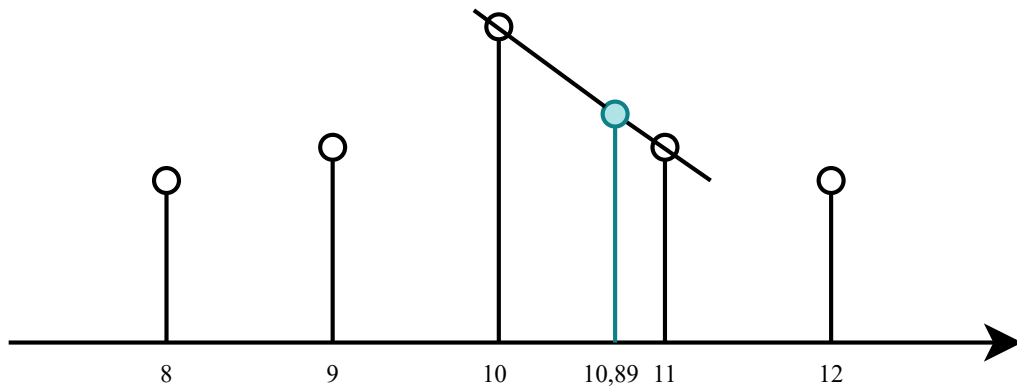
Figura 7.4 – *Buffer* com dois ponteiros



Fonte: Produção do autor.

Porém, como dito anteriormente, o atraso é modulado, e pode ter valores fracionários. Como o vetor só possui índices com valores inteiros, deve-se utilizar interpolações para não gerar ruídos no áudio. A interpolação mais utilizada é a linear (PIRKLE, 2013).

Figura 7.5 – Interpolação linear



Fonte: Adaptado de Pirkle (2013).

Como evidenciado na figura a seguir, caso deseja-se processar a amostra 10,89, deve-se utilizar uma interpolação linear, que é uma média ponderada entre a amostra passada e a seguinte. A equação para calcular essa média é dada a seguir (PIRKLE, 2013).

$$I[n] = (1 - frac) * x[n] + frac * x[n - 1] \quad (7.10)$$

No caso da figura 7.5, o valor "frac" é $(11 - 10,89)$.

7.1 OSCILADOR - LFO

Para variar o atraso, utiliza-se um oscilador de baixa frequência. Comercialmente, esse oscilador possui modulação senoidal, triangular ou randômica e é feito utilizando circuitos análogos, como gerador de onda quadrada, um integrador e etc. Para este projeto optou-se por gerar o oscilador via *software*, com modulação senoidal. A técnica escolhida é chamada de DDS - *Direct Digital Synthesizer* (WELCH; WRIGHT; MORROW, 2006).

A equação a seguir demonstra a modulação desejada para ser implantada em tempo real.

$$w(t) = A \sin(2\pi ft) \quad (7.11)$$

O parâmetro "A" da equação anterior refere-se a amplitude da modulação, e em pedais comerciais refere-se ao parâmetro *depth* que controla a amplitude da modulação. Essa amplitude interfere na quantidade de atraso injetado no sinal. A variável "f" representa a frequência desejada do oscilador e "t" é o vetor de tempo.

Porém, como deseja-se implementar em um microcontrolador, deve-se transformar a equação (7.11) para o tempo discreto. A equação a seguir apresenta a equação (7.11) em tempo discreto.

$$w[n] = \text{Asen}[2\pi f n T_s] = \text{Asen}[n(2\pi \frac{f}{F_s})] = \text{Asen}[n\phi_{inc}] = \text{Asen}(\phi_{acc}) \quad (7.12)$$

Onde t foi substituído por nT_s . O termo $2\pi \frac{f}{F_s}$ é chamado de incremento de fase e é alterado a cada T_s segundos. O argumento da função seno " $n\phi_{inc}$ " é chamado de acumulador de fase, e é gerado somando o incremento de fase com o valor anterior do acumulador de fase, como é demonstrado na equação a seguir.

$$\phi_{acc} = \phi_{acc} + \phi_{inc} \quad (7.13)$$

Como a função seno é periódica com período de 2π , então o acumulador de fase deve ter valor máximo de 2π , após esse valor, o acumulador é zerado e fica pronto para o próximo período da onda gerada.

A seguir, é apresentado um algoritmo simples para a implementação do LFO.

Algoritmo 3: Oscilador - LFO com modulação senoidal

Entrada: Frequência desejada em Hz, amplitude do atraso "A" em segundos

Saída: Valor da senoide

início

$$\phi_{inc} = \frac{2\pi f}{F_s};$$

$$\phi_{acc} = \phi_{acc} + \phi_{inc};$$

$$y = \text{Asen}(\phi_{acc});$$

se $\phi_{acc} > 2\pi$ **então**

$$\quad \phi_{acc} = \phi_{acc} - 2\pi;$$

fim

fim

Deve-se salientar que o algoritmo anterior deve ser realizado a cada interrupção, ou seja, a cada período de amostragem. A função seno demonstrada no algoritmo não deve ser implementada com a biblioteca "*math.h*" da linguagem C. Os processadores ARM possuem funções próprias para processamento de sinais com a biblioteca "*CMSIS-DSP*". Essa biblioteca deve ser baixada na *IDE - Integrated Development Environment* desejada. As funções trigonométricas, por padrão na linguagem C, são muito ineficientes.

Para corrigir esse problema, pode-se implementar uma *look-up-table*, que nada mais é do que um vetor com vários valores previamente armazenados da função trigonométrica. Então, quando deve-se performar o código da função, o algoritmo apenas pega o valor tabelado e faz a operação. Para este projeto, optou-se por utilizar a forma mais eficiente de processamento envolvendo os processadores ARM, que é a utilização da *CMSIS-DSP*. A função "*arm_sin_f32*" dessa biblioteca é uma função que trabalha com ponto flutuante e 32bits. A função utiliza uma

tabela com 512 amostras para a senoide, e utiliza uma interpolação linear para calcular índices do vetor que são não inteiros (DSP, 2021).

7.2 SÍNTESE DO CAPÍTULO

Este capítulo apresentou o caso especial de um filtro pente com atrasos variáveis. Foi apresentado o equacionamento necessário, o efeito produzido devido as alterações do mesmo, algoritmos e exemplos para auxiliar no desenvolvimento do mesmo. Os resultados desse efeito serão apresentados na seção 9.

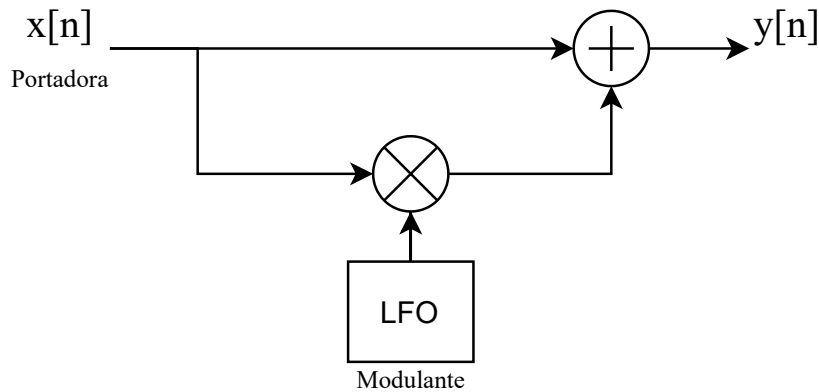
8 TREMOLO

As duas últimas seções trataram a respeito da modulação de fase. Outro tipo de modulação muito utilizado em áudio é a modulação em amplitude, conhecida pela sigla "AM - *Amplitude Modulation*". Em aplicações de radio frequência, o sinal de áudio é multiplicado por uma portadora (geralmente uma onda cossenoidal) de frequência muito alta, o objetivo é tornar o tamanho físico das antenas viáveis pra construção. Em aplicações de áudio, esse efeito consiste em multiplicar o sinal de áudio por um LFO. A equação a seguir representa essa ideia.

$$y[n] = x[n] \times (1 + \alpha \cos(2\pi \frac{f}{F_s} n)) \quad (8.1)$$

E o diagrama em blocos é apresentado a seguir.

Figura 8.1 – Diagrama de blocos para efeito tremolo



Fonte: Produção do autor.

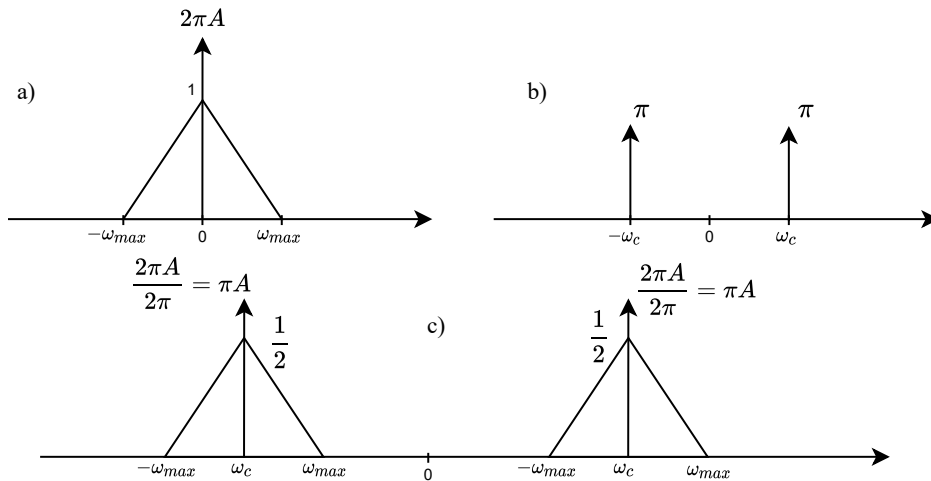
Onde x é o sinal de áudio e o termo entre parênteses é chamado de modulante, denominado por "c". A transformada de *Fourier* do produto da equação anterior é:

$$Y(j\omega) = \frac{1}{2\pi} X(j\omega) * C(j\omega) \quad (8.2)$$

A multiplicação no domínio do tempo é o mesmo que a convolução no domínio da frequência dividido por 2π . A figura a seguir apresenta as formas de onda da multiplicação. A figura 8.2a, apresenta uma transformada para o sinal modulante, com um nível DC de A. A transformada, é um impulso, referente ao nível DC em conjunto com uma forma de onda genérica, referente a forma de onda da portadora. A figura 8.2b, apresenta a transformada de *Fourier* da portadora, que é um cosseno.

A convolução dos sinais, no domínio da frequência, é a multiplicação entre as figuras 8.2a e 8.2b, resultando na figura 8.2c. Percebe-se o sinal duplicado no espectro, pois a multiplicação por um impulso é trivial. A amplitude da onda genérica é $\frac{\pi}{2\pi} = \frac{1}{2}$ e a amplitude do impulso é dado na figura. O termo 2π no denominador é referente a equação (8.2).

Figura 8.2 – Transformada de *Fourier* dos sinais



Fonte: Produção do autor.

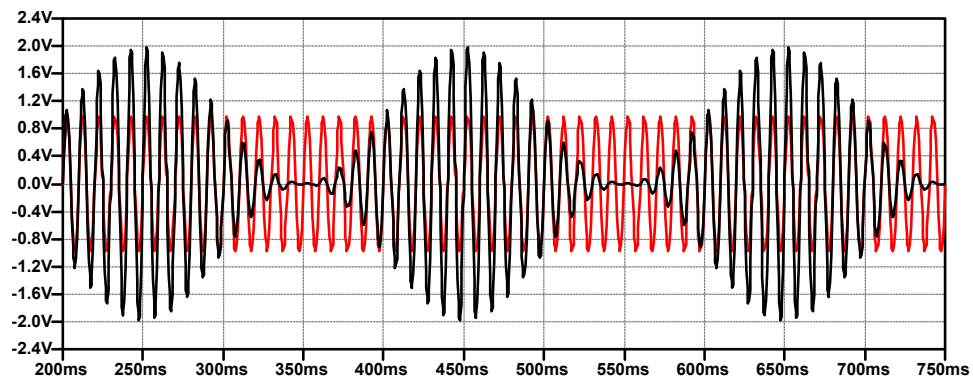
8.1 SIMULAÇÃO

Para aplicações de áudio, a portadora é o sinal de áudio e a modulante é uma senoide ou cosseno com baixa frequência. Para fins de simulação, como portadora foi escolhido um cosseno com frequência de $100Hz$ e amplitude unitária. A modulante foi escolhido como uma senoide de frequência $5Hz$ e amplitude unitária.

Na figura 8.3 pode-se observar o sinal mixado (onda com cor preta), em conjunto com a portadora, em vermelho. Da equação (8.1), percebe-se que os valores máximos, ocorrem quando as senoides possuem valor unitário, dessa forma, os valores variam entre $+1$ até $-1V$.

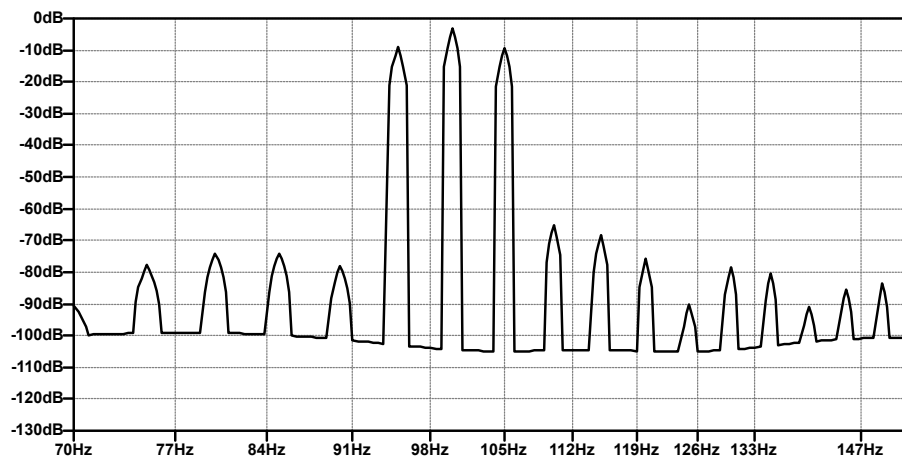
As figuras 8.4 e 8.5 demonstram o espectro de frequências para o sinal modulado com e sem *zoom*, respectivamente. Das figuras, pode-se perceber o sinal duplicado na frequência da modulante ($5Hz$) variando entre $95Hz$ e $100Hz$.

Figura 8.3 – Forma de onda para o sinal modulado e a portadora



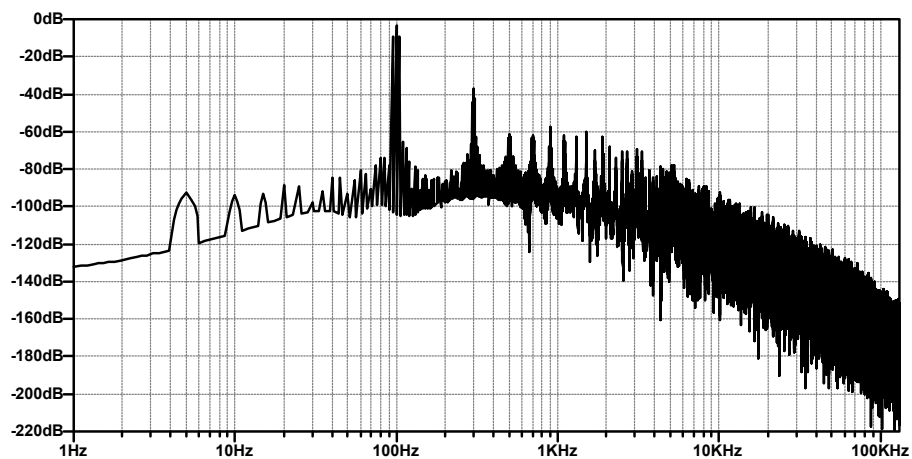
Fonte: Produção do autor.

Figura 8.4 – Forma de onda para a FFT do sinal modulado com *zoom*



Fonte: Produção do autor.

Figura 8.5 – Forma de onda para a FFT do sinal modulado sem *zoom*



Fonte: Produção do autor.

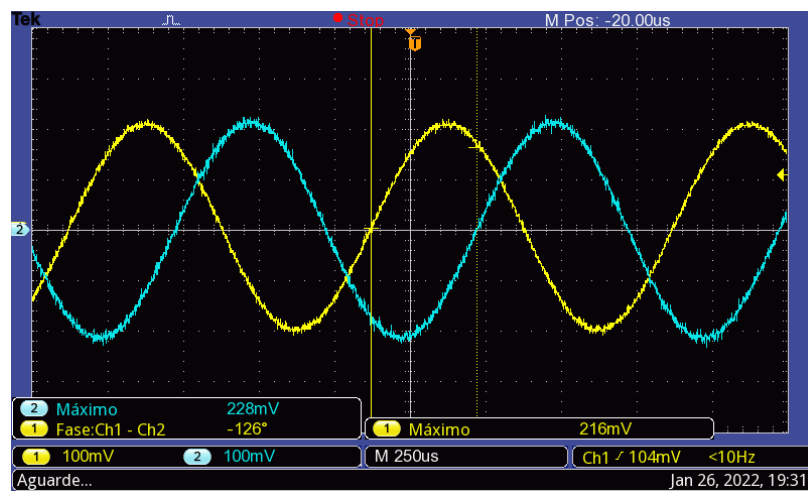
8.2 SÍNTESE DO CAPÍTULO

Nesse capítulo foi apresentado o contexto teórico por trás do efeito de modulação em amplitude *tremolo*. O efeito consiste basicamente de um oscilador de baixas frequências que foi apresentado inicialmente na seção 7.1. Os resultados em tempo real serão apresentados na seção a seguir.

9 RESULTADOS

Esta seção será dedicada a apresentação dos resultados obtidos em tempo real do pedal proposto. O sinal de entrada será uma senoide de 1kHz e 200mV de pico. Além disso, resultados sonoros podem ser encontrados em <<https://github.com/Emersonandrey11/Digital-Guitar-Stompbox>>.

Figura 9.1 – Sinais de entrada e saída



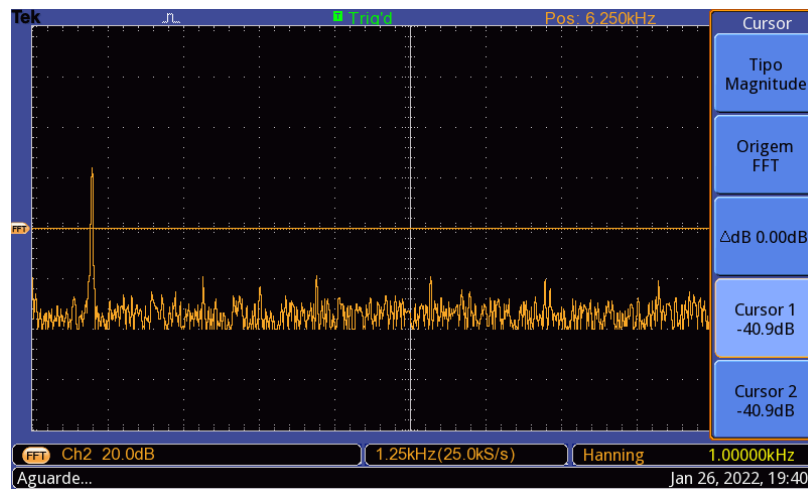
Fonte: Produção do autor.

A saída está atrasada da entrada, como demonstra o cursor, em aproximadamente $350\mu\text{s}$. É um atraso imperceptível ao ouvido humano e surge através dos filtros implementados e do processamento. O ganho é $\frac{228\text{m}}{216\text{m}} = 1,055$, próximo ao projetado.

A figura a seguir apresenta a transformada de *Fourier* do sinal de saída, e demonstra que o circuito tem baixa distorção harmônica.

De forma geral, o circuito apresenta uma boa qualidade de processamento, tanto no leiaute da placa de circuito impresso, quanto no processamento dos algoritmos, de acordo com essas duas figuras.

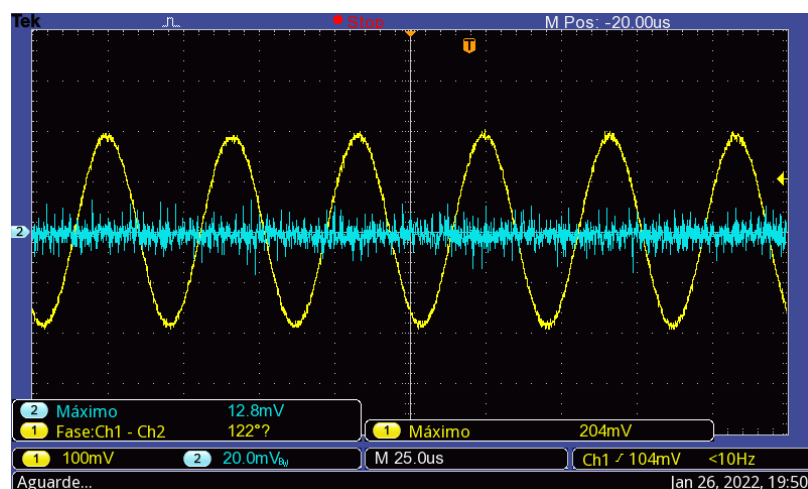
Figura 9.2 – FFT do sinal de saída



Fonte: Produção do autor.

A figura a seguir apresenta a mesma relação de entrada saída. Porém, a onda em amarelo possui 24kHz . Essa figura demonstra que para frequências acima de 8kHz , há uma elevada atenuação. Com um sinal de 24kHz , praticamente não há sinal na saída com o canal em 20mV por divisão. Com isso, o sinal de saída é praticamente da mesma amplitude que algum ruído captado pelo osciloscópio, ou seja, é desprezível. Dessa forma, a escolha em diminuir a ordem do filtro *antialiasing* para cinco é aceitável.

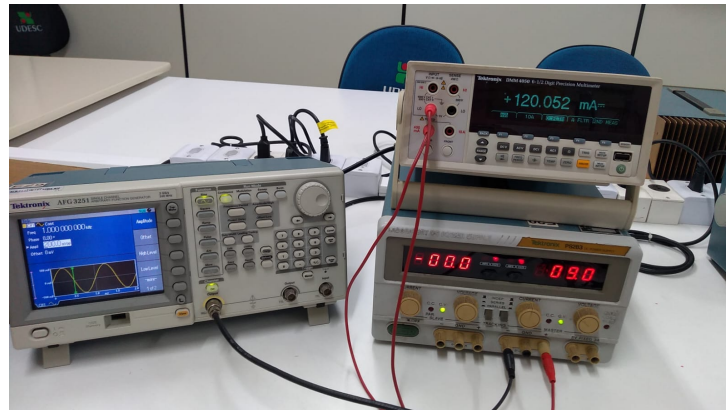
Figura 9.3 – Atuação dos filtros analógicos



Fonte: Produção do autor.

A figura a seguir apresenta uma foto retirada da bancada experimental. O sistema é alimentado com uma fonte de bancada com 9V. Em série com o sistema, é adicionado um amperímetro digital para calcular o consumo do projeto. O consumo total foi de $120,052\text{mA}$ que demonstra que o regulador escolhido mostra-se mais que suficiente para suportar a corrente necessária.

Figura 9.4 – Corrente consumida

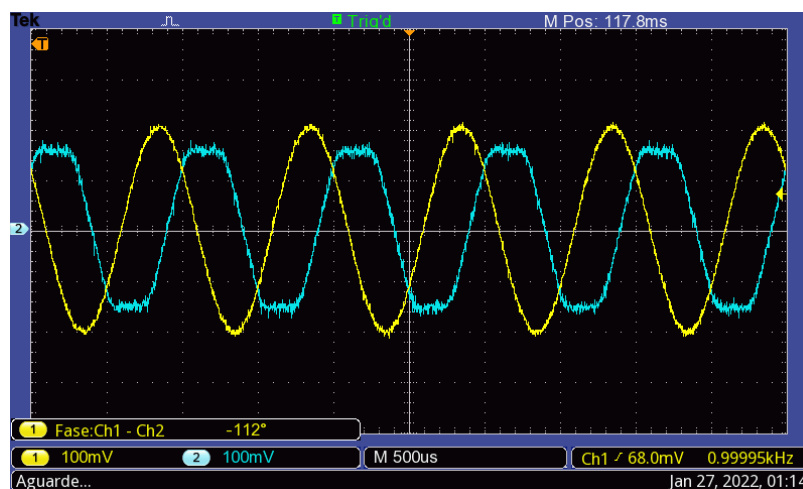


Fonte: Produção do autor.

9.1 DISTORÇÃO

A figura a seguir apresenta o algoritmo de distorção, utilizando *hard clipping* (equação (5.2)).

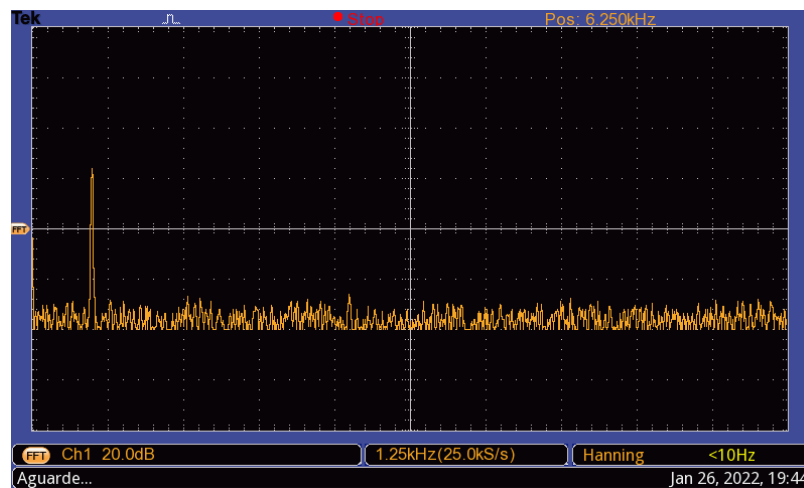
Figura 9.5 – Algoritmo de Distorção



Fonte: Produção do autor.

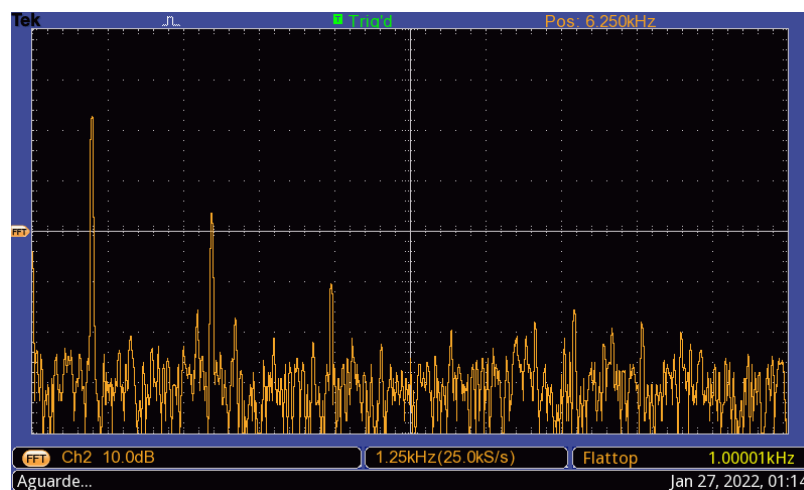
As duas figuras a seguir demonstram a comparação entre o sinal de entrada sem efeito e o sinal de saída com efeito de distorção. A figura 9.6 mostra a transformada de *Fourier* da entrada e a figura 9.7 mostra a transformada da saída.

Figura 9.6 – FFT do sinal de entrada



Fonte: Produção do autor.

Figura 9.7 – FFT do algoritmo de distorção



Fonte: Produção do autor.

Ao mudar o ponto de corte, é introduzido mais harmônicas ao sinal de saída.

9.2 DELAY

Para demonstrar os resultados obtidos desse efeito, foi utilizado um pulso de $1Hz$ com 5% de *duty cycle* (onda em amarelo). A forma de onda ideal era um impulso, porém como não há implementação prática do mesmo foi escolhido essa forma de onda, para que a energia do sistema se dissipe antes que o próximo pulso aconteça. A equação utilizada será repetida aqui para fins de visualização.

$$y[n] = 1 \times x[n] + 0,7 \times x[n - 9600] \quad (9.1)$$

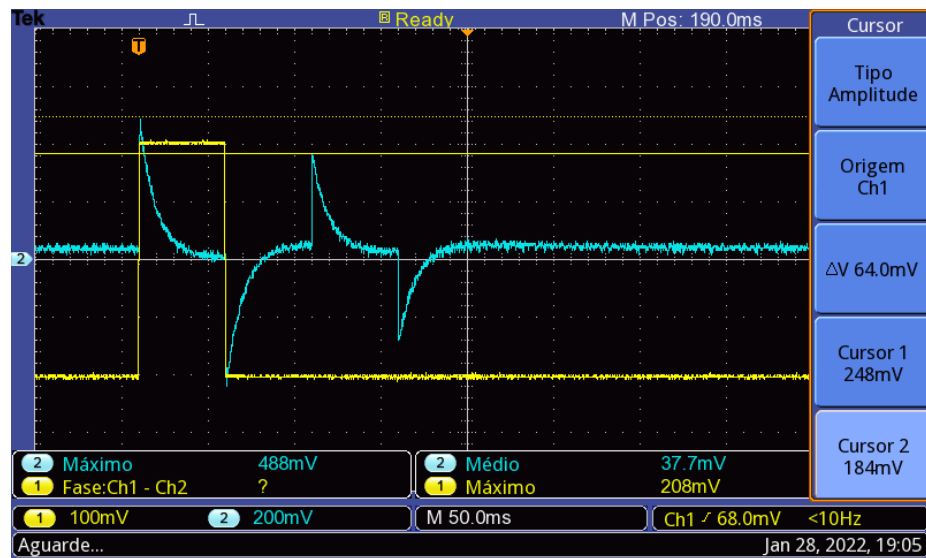
Onde o valor para o coeficiente *dry*, descrito na seção 6, foi utilizando como um, e o coeficiente *wet* foi utilizado como 0,7. O valor de 9600 representa um atraso apenas para fins de demonstração, e possui um valor de atraso de $\frac{9600}{f_s} = 100ms$, com $f_s = 96kHz$.

Outro fator importante é que, devido a baixa frequência do pulso (apenas $1Hz$), a forma de onda de entrada foi inserida diretamente no conversor AD, o estágio de filtro foi retirado, pois como descrito na seção 3, há um filtro passa-alta na entrada e o mesmo acaba atenuando a resposta do sistema para baixas frequências.

Então, a aquisição de dados para este efeito foi feita inserindo o pulso no conversor AD, em seguida o conversor envia o valor digital para o MCU. O MCU então processa os dados, de acordo com o algoritmo desenvolvido, e envia para o conversor DA. Tal processo pode ser visualizado na figura a seguir, onde o canal um (onda em amarelo) representa o pulso de entrada, e o canal dois (onda em azul) apresenta a resposta a esse pulso. A forma de onda em azul, da figura 9.9, se mostrou um pouco ruidosa na bancada, e foi utilizado uma média de 16 amostras no osciloscópio, para demonstrar, sem ruídos, a forma de onda.

As fontes de ruídos presenciadas, são provavelmente a dificuldade de medição do osciloscópio devido a baixa frequência da onda e a amostragem do sinal sem filtro *antialiasing*.

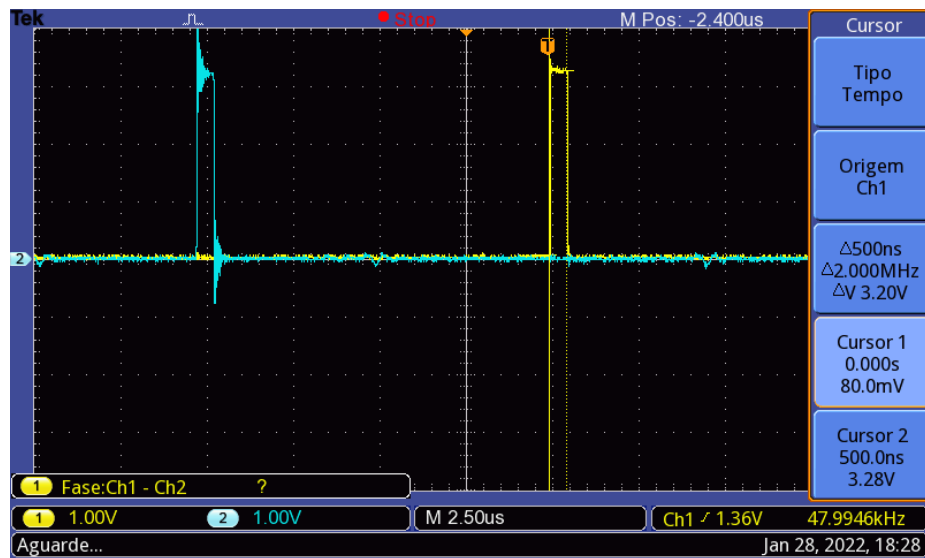
Como pode-se observar da figura a seguir, a onda em azul tem uma cópia atrasada de $100ms$, e amplitude de 0,7 vezes a tensão de entrada. A amplitude da tensão em azul deveria ser aproximadamente 1,7 vezes maior o que a da entrada, como demonstrado pela equação (6.4). Porém, como na prática tem-se um degrau ao invés de um impulso aplicado, há alguma diferença nos valores, além disso, o osciloscópio está trabalhando com uma frequência de $1Hz$ e $100mV$, e é normal o aparelho captar algum ruído que interfira na captação.

Figura 9.8 – Resposta ao degrau - *Delay*

Fonte: Produção do autor.

A figura a seguir demonstra o conceito de *double buffer* para a rotina dess efeito. A onda em azul representa o processamento da primeira interrupção e a onda em amarelo a segunda interrupção. O tempo de cada interrupção foi medido como $500ns$ e apresenta um atraso imperceptível.

Figura 9.9 – Double buffer - Delay



Fonte: Produção do autor.

9.3 FLANGER

Para a demonstração de resultados deste efeito foi utilizado o mesmo princípio da seção anterior. Como sinal de entrada, foi utilizado um pulso, porém com frequência de 1Hz e *duty cycle* de 5% . Dessa forma, pode-se utilizar uma frequência maior, e mesmo assim o pulso terá tempo suficiente para se dissipar antes que o próximo pulso aconteça. A equação implementado no algoritmo pode ser visualizada novamente a seguir.

$$y[n] = x[n] + \alpha x[n - LFO] \quad (9.2)$$

A equação do LFO é dada a seguir.

$$LFO = D + \beta \times \sin\left(\frac{2\pi n f}{f_s}\right) \quad (9.3)$$

O valor do parâmetro "D" foi escolhido arbitrariamente como 100ms , e o valor do β foi escolhido como 30ms . Dessa forma, o sinal oscilará de forma senoidal com atraso máximo de 130ms e mínimo de 70ms . Estes valores não são práticos, porém evidenciam de uma forma melhor os efeitos no osciloscópio.

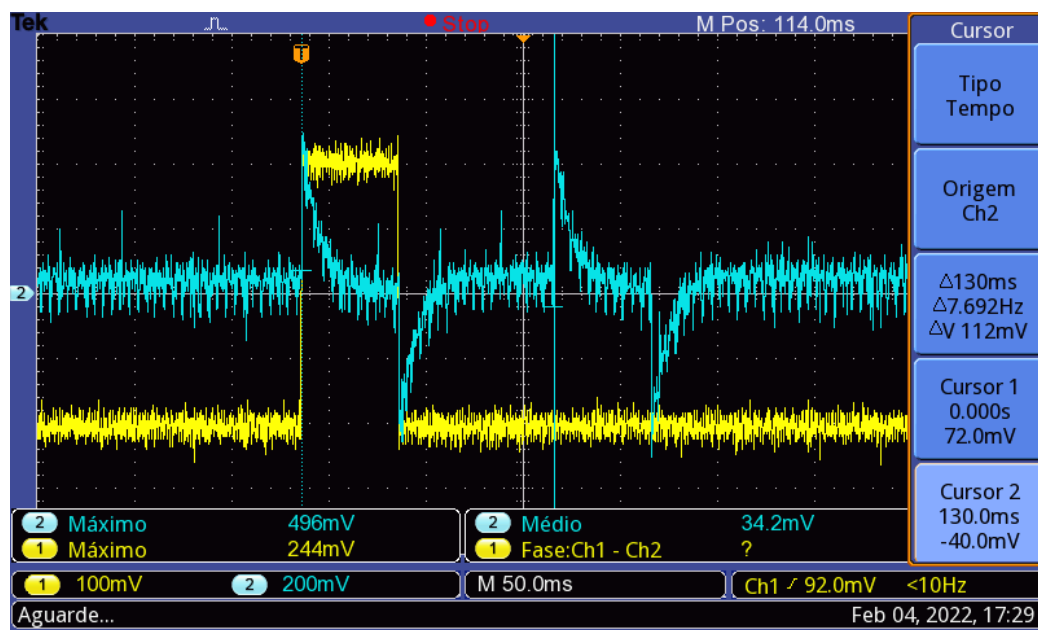
O parâmetro "f" que implica na frequência de oscilação foi escolhido de forma arbitrária como $0,8\text{Hz}$ e seu comportamento é impossível de ser captados por fotos. Dessa forma, um

vídeo foi capturado como forma de aquisição de dados e está contido no link para o *GitHub* disponibilizado.

As figuras a seguir demonstram a resposta ao pulso. Novamente, percebe-se um ruído no sinal e as causas são as mesmas da seção anterior. Neste efeito não há a possibilidade de usar uma média no sinal pois o mesmo varia senoidalmente, e a média utilizada pelo osciloscópio possui uma resposta lenta.

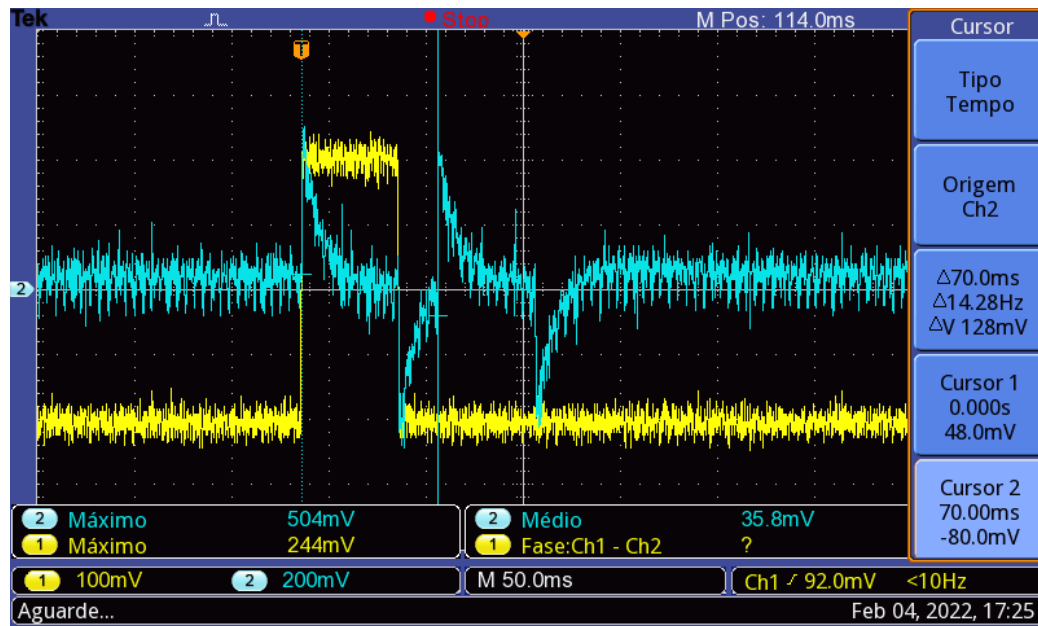
A figura 9.10 apresenta o resposta do filtro pente com um atraso variável máximo, e a figura 9.12 representa o atraso mínimo. O segundo pico, que representa o atraso varia, em tempo real entre os valores máximos e mínimos. Um vídeo demonstrando essa variação em tempo real pode ser encontrado em <<https://github.com/Emersonandrey11/Digital-Guitar-Stompbox>>.

Figura 9.10 – Resposta ao degrau - *Flanger* com máxima variação



Fonte: Produção do autor.

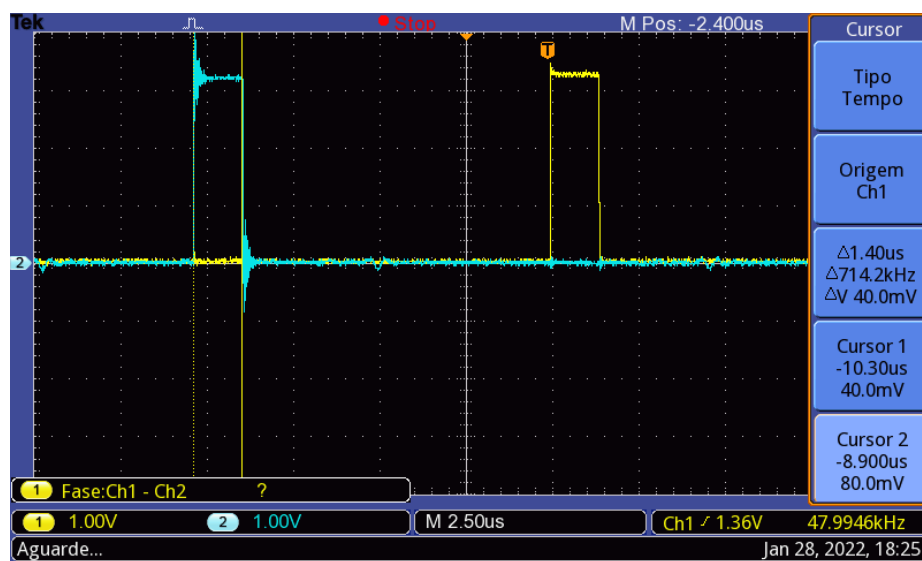
Figura 9.11 – Resposta ao degrau - *Flanger* com mínima variação



Fonte: Produção do autor.

A figura a seguir demonstra técnica de *double buffer* para a rotina *flanger*. Da figura, pode-se observar que a rotina leva um tempo de $1,4\mu\text{s}$ para ser executada a cada interrupção.

Figura 9.12 – *Double buffer* para a rotina *Flanger*



Fonte: Produção do autor.

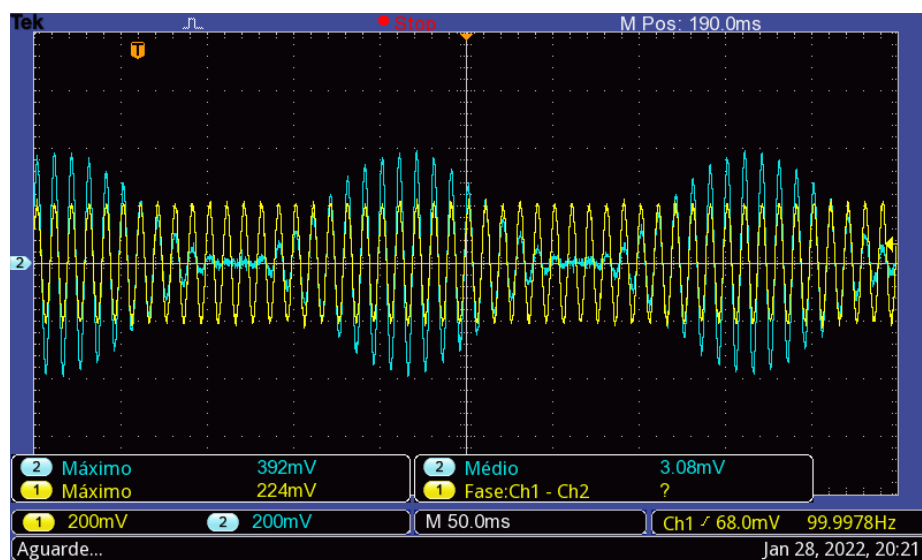
9.4 TREMOLO

Para este efeito foi aplicado uma portadora senoidal, simulando um sinal de áudio com 100Hz , e uma modulante, representado o sinal do oscilador de baixas frequências de 5Hz .

A figura 9.13 tem como canal um (amarelo) a portadora, e o canal dois (azul) o sinal modulado em amplitude. A figura 9.14 apresenta o espectro do sinal modulado. Pode-se observar que existe a harmônica referente a frequência da onda portadora em 100Hz , e também há as harmônicas referentes a modulante em 95Hz e 105Hz .

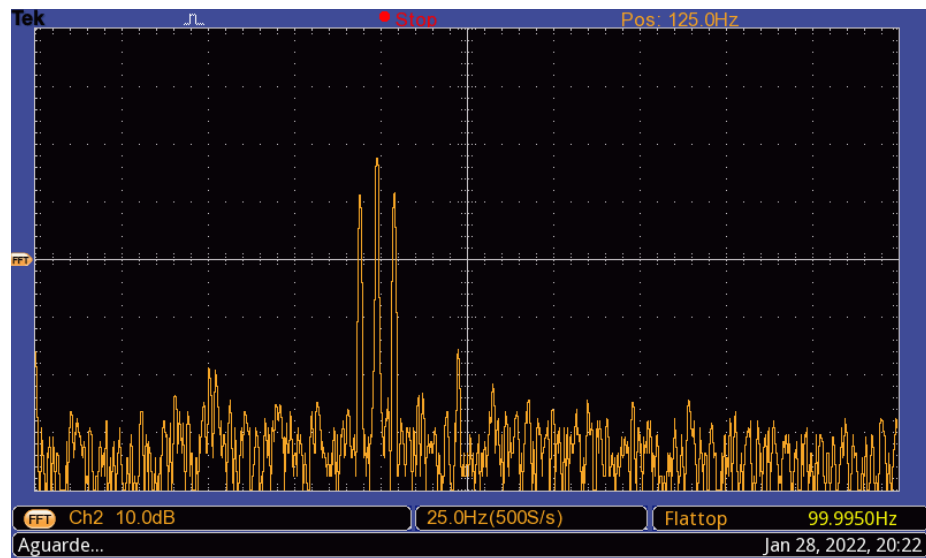
A figura 9.15 representa o conceito de *double buffer* para a rotina do algoritmo *tremolo*. Cada rotina apresenta um tempo de $1,10\mu\text{s}$.

Figura 9.13 – Efeito tremolo no domínio do tempo



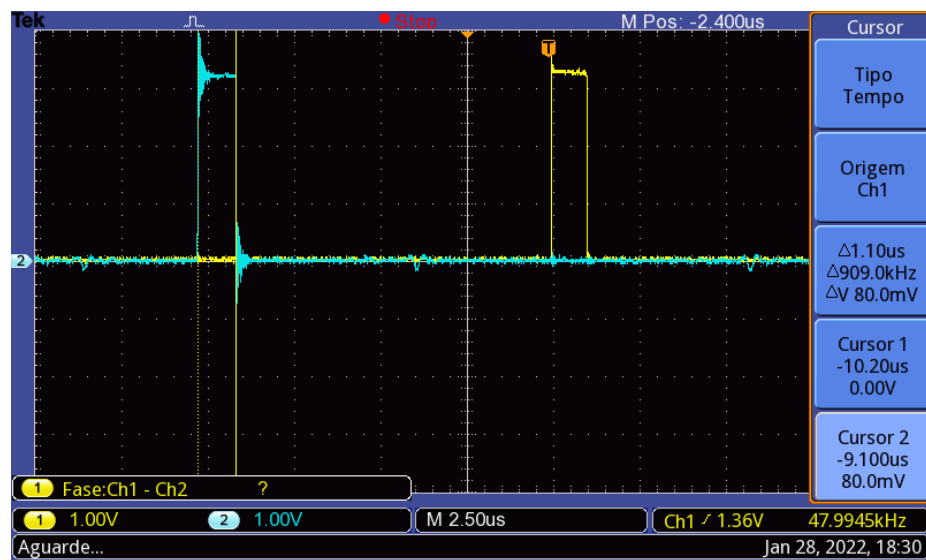
Fonte: Produção do autor.

Figura 9.14 – Efeito Tremolo no domínio da frequência



Fonte: Produção do autor.

Figura 9.15 – *Double buffer* para o efeito Tremolo



Fonte: Produção do autor.

9.5 SÍNTESE DO CAPÍTULO

Esse capítulo procurou apresentar um efeito de modulação em amplitude aplicado a áudio. Através da figura anterior, pode-se observar que o tempo da rotina é de apenas $1,10\mu s$, tornando o mesmo aplicável em tempo real e simulando digitalmente um efeito muito popular no meio musical. Conclui-se que a função trigonométrica "*arm_sin_f32*" desempenha um bom papel para algoritmos com modulação em amplitude.

10 CONCLUSÃO

Esse trabalho de conclusão de curso apresentou um processamento em tempo real de áudio utilizando um ARM córtex M4. Os capítulos desenvolvidos procuraram demonstrar uma forma eficiente de transmitir dados de um instrumento de corda digitalmente. Circuitos analógicos de condicionamento de sinais foram projetados e testados em bancada. Um atraso de $350\mu s$ entre entrada e saída foi observado, porém o mesmo não é perceptível. Algoritmos de processamento de sinais foram desenvolvidos, envolvendo atrasos e modulações. O tempo de rotina de cada um deles foi apresentado na prática. O maior tempo entre eles é da rotina do efeito *Flanger* com atraso de $1,40\mu s$, o que não implica em algo incômodo. O pedal não apresenta ruídos e não introduz harmônicas no sinal, como esperado.

Conclui-se que o microcontrolador utilizado possui excelentes aplicações para áudio e capacidade de armazenamento para até $300ms$ de áudio, que em muitos casos, é maior do que o suficiente.

Para trabalhos futuros, os seguintes tópicos são sugeridos:

- Modelo digital do algoritmo de distorção, através de simulação SPICE ou análise de esquemáticos, pode-se ter uma resposta mais fiel a efeitos de distorção;
- Melhoria na placa de circuito impresso. O projeto utiliza uma placa de desenvolvimento, uma melhora seria desenvolver a própria placa com componentes SMD. Essa ação, diminuiria significativamente o consumo do sistema.
- Uso de um regulador chaveado para a diminuição do consumo
- Uso de um circuito integrado contendo filtros *antialiasing*, esses circuitos são mais caros, em contrapartida, não inserem atraso significativo no sinal.

REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, G. C.

Projeto de unidade de efeito sono para guitarras — Trabalho de Conclusão de Curso – Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná, Curitiba, Brasil, 2018.

ARAYA, T.; SUYAMA, A. **Sound effector capable of imparting plural sound effects like distortion and other effects**. [S.l.]: U.S. Patent no. 5570424, 1996.

BAKER, B. C. **Anti-Aliasing, Analog Filters for Data Acquisition Systems**. 1. ed. [S.l.]: Microchip Technology Inc., 1999.

BALLOU, G. **Handbook for Sound Engineers**. 4. ed. Nova York, EUA: Focal Press - Elsevier, 2008. 1778 p.

BARBOUR, E. The cool sound of tubes [vacuum tube musical applications]. **IEEE Spectrum**, 1998. v. 35, n. 8, p. 24–35, 1998.

BERTOLDI, B.

Estudo de um amplificador classe D destinado à guitarra elétrica — Trabalho de conclusão de curso (Bacharelado em Engenharia Elétrica – Área: Eletrônica de Potência) - Universidade do Estado de Santa Catarina, Joinville, Brasil, 2016.

BOSS. **DS - 1X Distorção**. 2020. Disponível em: <<https://www.boss.info/br/products/ds-1x/features/>>. Acessado em: 01/08/2020.

BUŚ, S.; JęDRZEJEWSKI, K. Realization of guitar audio effects using methods of digital signal processing . In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2015**. SPIE, 2015. v. 9662, p. 589 – 595. Disponível em: <<https://doi.org/10.1117/12.2205439>>.

CLAYTON, G.; WINDER, S. **Operational Amplifiers**. 5. ed. Oxford, Inglaterra: Newnes - Elsevier, 2003. 386 p.

COSTA, E. A. de. **Virtualização da rede Zobel para aplicação em amplificadores de áudio classe de D**. Dissertação (Mestrado) — Universidade do Estado de Santa Catarina, Joinville, Brasil, 2020.

DAILEY, D. J. **Electronics for Guitarists**. 1. ed. Nova York, EUA: Springer, 2011. 290 p.

DINIZ, P. S. R.; SILVA, E. A. B. da; NETTO, S. L. **Digital Signal Processing**. 2010. ed. [S.l.]: Cambridge University Press, 2005.

DSP, C. **CMSIS DSP Software Library**. 2021. Disponível em: <https://www.keil.com/pack/doc/CMSIS/DSP/html/group__sin.html>. Acessado em: 26/03/2021.

FENDER. **Vintera '50S Stratocaster**. 2020. Disponível em: <<https://shop.fender.com/en/intl/electric-guitars/stratocaster/vintera-50s-stratocaster-modified/0149962304.html>>. Acessado em: 26/06/2020.

GIBSON. **Les Paul Standard**. 2020. Disponível em: <<https://www.gibson.com/Guitar/CUSQKW566/60th-Anniversary-1960-Les-Paul-Standard/V1-Deep-Cherry-Sunburst>>. Acesso em: 26/06/2020.

HUNTER, M. **A Technical Analysis and Exploration of Guitar Pickups and Electronics**. [S.l.]: Senior Projects Spring, 2018. 359 p.

HUOVILAINEN, A. Enhanced digital models for analog modulation effects. In: **8th Int. Conference on Digital Audio Effects (DAFX-05)**. [S.l.: s.n.], 2005. p. 155–160.

INSTRUMENTS, T. **2VRMS DirectPath™, 112/106/100dB Audio Stereo DAC with 32-bit, 384kHz PCM Interface**. 2012. Disponível em: <https://www.ti.com/lit/ds/symlink/pcm5102.pdf?ts=1616748007611&ref_url=https%253A%252F%252Fwww.google.com%252F>. Acesso em: 26/03/2021.

INSTRUMENTS, T. **PCM1802 Single-Ended Analog-Input 24-Bit, 96-kHz Stereo A/D Converter**. 2016. Disponível em: <https://www.ti.com/lit/ds/symlink/pcm1802.pdf?ts=1596331675794&ref_url=https%253A%252F%252Fwww.google.com%252F>. Acesso em: 01/08/2020.

JUNGSMANN, T. **Theoretical and practical studies on the behaviour of electric guitars pickups**. Dissertação (Mestrado) — Helsinki University of Technology, Espoo, Finlândia, 1994.

KARKI, J. **Active Low-Pass Filter Design**. Application report - sloa049b. [S.l.]: Texas Instruments, 2002.

KESTER, W. **Data Conversion Handbook**. 1. ed. [S.l.]: Newnes - Elsevier, 2005.

KESTER, W. **Taking the Mystery out of the Infamous Formula, SNR = 6.02N + 1.76dB, and Why You Should Care**. 2008. Disponível em: <<https://www.analog.com/media/en/training-seminars/tutorials/MT-001.pdf>>. Acesso em: 23/11/2021.

LANG, I.

Digital Guitar Effects Pedal — TCC (Graduação) - California Polytechnic State University, San Luis Obispo, Estados Unidos, 2018.

MILLS, S. E. **LO IMPEDANCE DUAL COIL BIFILAR MAGNETIC PICKUP**. United States Patent - US 9,524,710 B2, 2016.

MOHAMAD, Z.; DIXON, S.; HARTE, C. Pickup position and plucking point estimation on an electric guitar. In: **2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2017. p. 651–655.

OLIVEIRA, T. C. d. A.; BARRETO, G.; PASQUAL, A. M. Review of digital emulation of vacuum-tube audio amplifiers and recent advances in related virtual analog models. **INFOCOMP Journal of Computer Science**, 2013. v. 12, n. 1, p. 10–23, Jun. 2013. Disponível em: <<https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/369>>.

ORFANIDIS, S. J. **Signal Processing**. 1. ed. Sem Local: Pearson Education, 2010. 795 p.

OVERTON, W. E. **Digital Circuit-Level Emulation of Transistor-Based Guitar Distortion Effects**. Dissertação (Mestrado) — Georgia Institute of Technology, Atlanta, Estados Unidos, 2006.

OWSINSKI, B. **Modulation effects: Flanging, Phase Shifting, and more**. 2021. Disponível em: <<https://www.uaudio.com/blog/modulation-effects-basics/>>. Acessado em: 23/11/2021.

PAIVA, R. C. D.; PAKARINEN, J.; VÄLIMÄKI, V. Acoustics and modeling of pickups. **J. Audio Eng. Soc.**, 2012. v. 60, n. 10, p. 768–782, 2012. Disponível em: <<http://www.aes.org/e-lib/browse.cfm?elib=16551>>.

PAKARINEN, J.; Yeh, D. T. A review of digital techniques for modeling vacuum-tube guitar amplifiers. **Computer Music Journal**, 2009. v. 33, n. 2, p. 85–100, 2009.

PAKARINEN, J.; YEH, D. T. A review of digital techniques for modeling vacuum-tube guitar amplifiers. **Computer Music Journal**, 2009. v. 33, n. 2, p. 85–100, 2009.

PIRKLE, W. C. **Designing Audio Effect Plugins in C++**. 2. ed. Nova York, EUA: Routledge, 2013. 703 p.

RAFFEL, C. Practical modeling of bucket-brigade device circuits. In: . [S.l.: s.n.], 2010.

RAMOS, A. **Eletrromagnetismo**. [S.l.]: Blucher, 2016. 354 p.

SCHAUMANN, R.; VALKENBURG, M. E. V. **Design of analog filters**. 1. ed. [S.l.]: Oxford University Press, 2001.

An Allpass Approach to Digital Phasing and Flanging. Disponível em: <<https://ccrma.stanford.edu/files/papers/stanm21.pdf>>.

STMICROELECTRONICS. 2021. Disponível em: <<https://www.st.com/en/microcontrollers-microprocessors/stm32f401.html>>. Acessado em: 26/03/2021.

STMICROELECTRONICS. **Interfacing an STM32L1xx microcontroller with an external I2S audio codec to play audio files**. 2014. Disponível em: <https://www.st.com/resource/en/application_note/dm00087544-interfacing-an-stm32l1xx-microcontroller-with-an-external-i2s-audio-codec-to-play-audio-files.pdf>. Acessado em: 01/08/2020.

STMICROELECTRONICS. **ARM® Cortex-M4 32b MCU+FPU, 105 DMIPS, 512KB Flash/96KB RAM, 11 TIMs, 1 ADC, 11 comm. interfaces**. [S.l.], 2015. Rev. 3.

TECHNOLOGY, S. J. H. Y. E. **PCM1802 Áudio Estéreo A/D Conversor Decodificador 24bit ADC 96kHz**. 2022. Disponível em: <<https://portuguese.alibaba.com/product-detail/a5-pcm1802-audio-stereo-a-d-converter-adc-decoder-24bit-96khz-digital-pcm-av-amplifier-player-board.html>>. Acessado em: 03/01/2022.

TIAN CHANG SU, C. Y. C. Research on electric parameters optimization of electric guitar pickup. In: **2nd International Conference on Systems, Computing, and Applications (SYSTCA 2018)**. Shenzhen, China: [s.n.], 2018. p. 69–75.

WEBSTER, T.; LENOST, G.; KLANG, M. The owl: an open source, programmable stage effects pedal. In: **KES Transactions on Innovation in Music**. [S.l.: s.n.], 2013. v. 1, n. 1, p. 147–157.

WELCH, T. B.; WRIGHT, C. H.; MORROW, M. G. **Real Time Digital Signal Processing: from MATLAB to C with the TMS320C6x DSK**. 1. ed. Nova York, EUA: Taylor Francis Group, 2006. 363 p.

WICKERT, M. A. Using the arm cortex-m4 and the cmsis-dsp library for teaching real-time dsp. In: **2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE)**. [S.l.: s.n.], 2015. p. 283–288.

WRIGHT, A.; VÄLIMÄKI, V. Neural modeling of phaser and flanging effects. **Journal of the Audio Engineering Society**, 2021. Audio Engineering Society, v. 69, n. 7-8, p. 517–529, jul. 2021. ISSN 1549-4950.

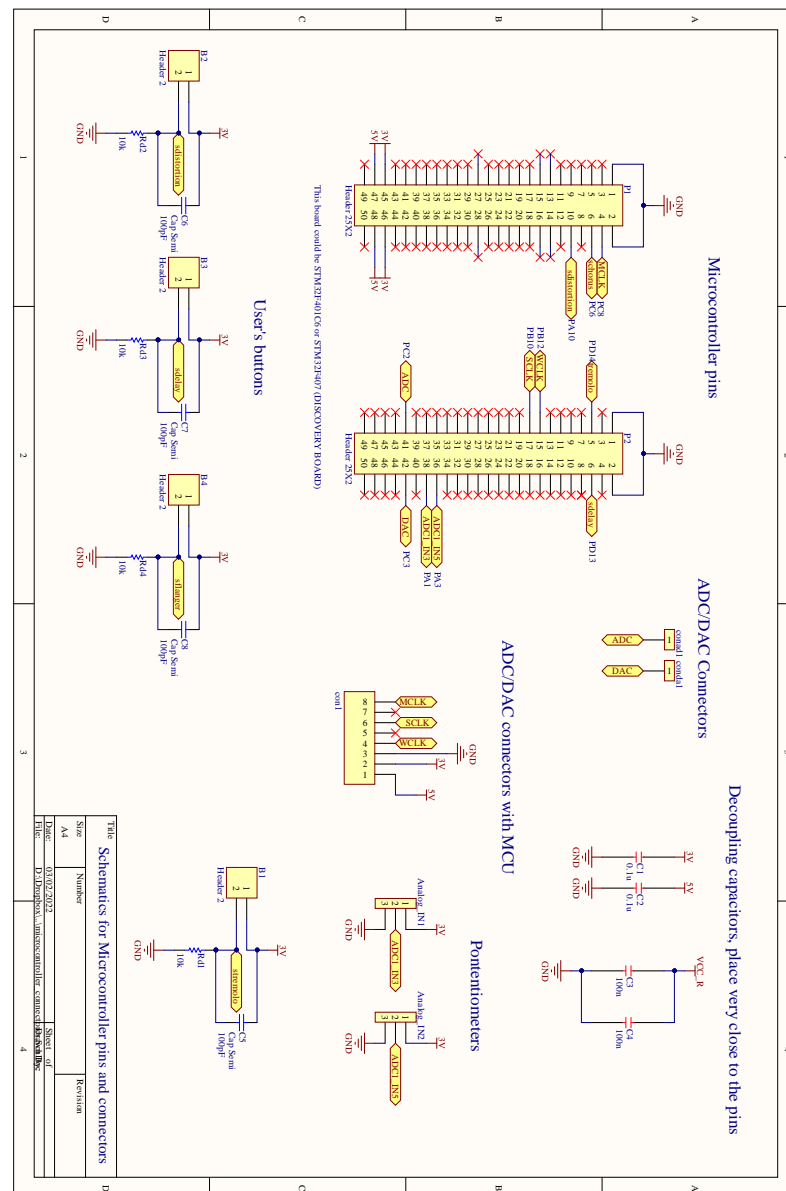
ZEKI, E. **Digital Modelling of Guitar Audio Effects**. Dissertação (Mestrado) — Middle East Technical University, Ankara, Turquia, 2015.

ZOLZER, U. **DAFX - Digital Audio Effects**. 1. ed. [S.l.]: John Wiley and Sons, Ltd, 2002.

ANEXO A – Esquemático PCI - Microcontrolador e pinagem

A seguir, encontram-se os esquemáticos e projeto da PCI. Os mesmos, podem ser encontrados *online* em: <<https://github.com/Emersonandrey11/Digital-Guitar-Stompbox>>. Nesse *link*, encontram-se também o código, arquivos esquemáticos e vídeos de demonstração.

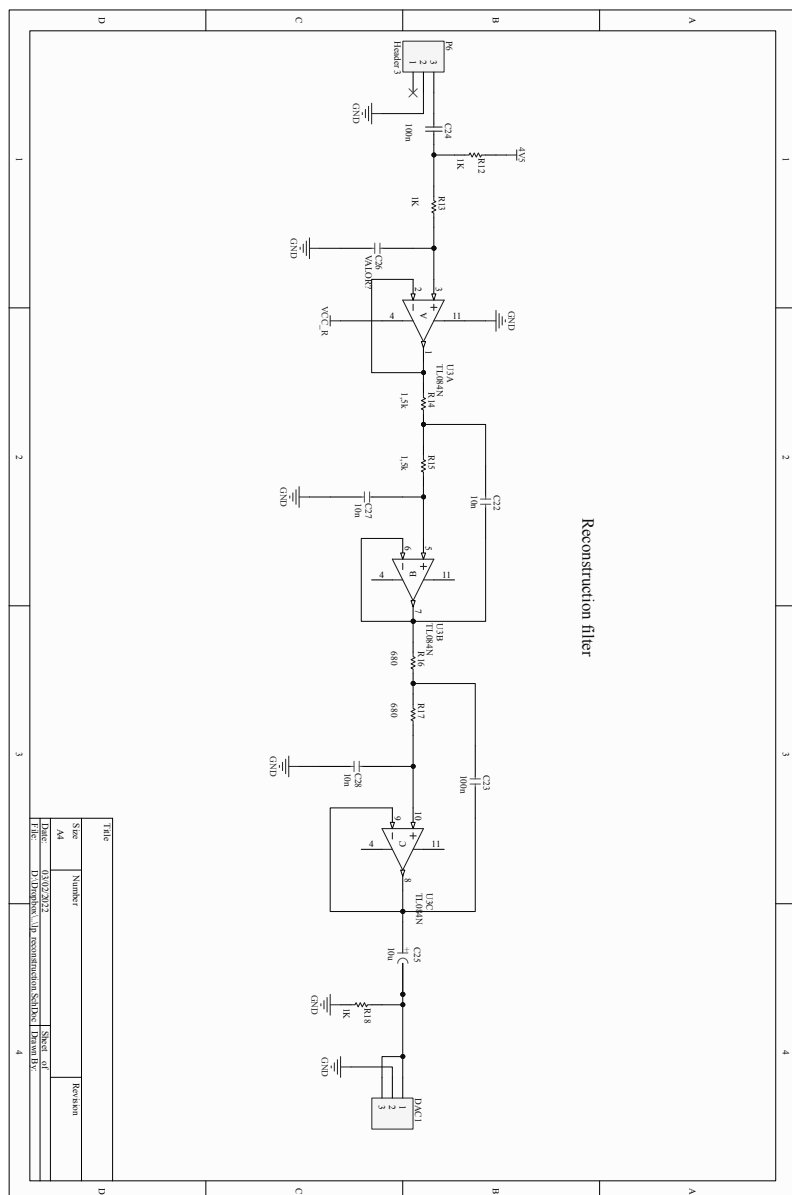
Figura A.1 – Pinagem do microcontrolador



Fonte: Produção do autor.

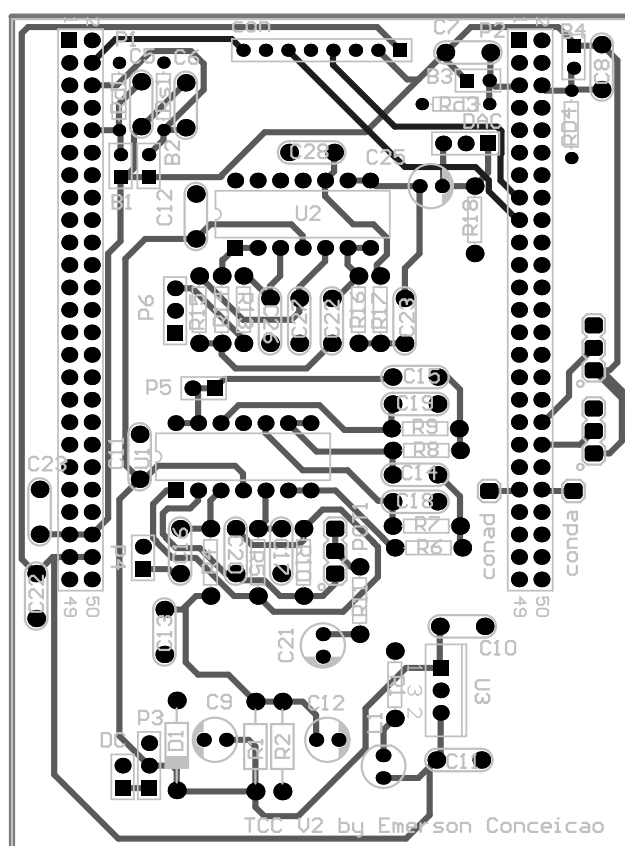
Fonte: Produção do autor.

Figura A.4 – Filtro de reconstrução



Fonte: Produção do autor.

Figura A.5 – Placa de circuito impresso



Fonte: Produção do autor.